Multiphysics and multiscale software frameworks : an annotated bibliography

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Multiphysics and Multiscale Software Frameworks: An Annotated Bibliography

Önder Babur, Tom Verhoeff and Mark van den Brand

15/01

ISSN 0926-4515

Reports are available at:
http://library.tue.nl/catalog/TUEPublication.csp?Language=dut&Type=ComputerScienceReports&Sort=Author&level=1 and
http://library.tue.nl/catalog/TUEPublication.csp?Language=dut&Type=ComputerScienceReports&Sort=Year&Level=1

Computer Science Reports 15-01
Eindhoven, April 2015

Multiphysics and Multiscale Software Frameworks: An Annotated Bibliography

Önder Babur, Tom Verhoeff, and Mark van den Brand

Eindhoven University of Technology, The Netherlands
{*o.babur, t.verhoeff, m.g.j.v.d.brand*}*@tue.nl*

Abstract

Multiphysics and multiscale modelling and simulation (MMS) is an emerging trend for the analysis and design of complex systems in many domains. As a result, there are an overwhelmingly large number of MMS software frameworks in the literature and market, while a comprehensive account of these is apparently missing. This paper presents an annotated bibliography of MMS software frameworks. A thorough bibliographic search in Scopus has been done, to find out the candidates in physical sciences, published from 2000 onwards. Further cross-references have been investigated to achieve a better coverage. The frameworks have been categorized according to their application areas, and annotated with respect to their main features regarding software integration/extension/coupling.

1 Introduction

Numerical simulations have been used in industry and academia for a few decades as a computational paradigm for research and development. Recently, and notably after the 2000s, it has become increasingly important that knowledge about various physical phenomena involving distinct space/time scales and scientific disciplines are integrated in an efficient way to promote further advancement [1]. This integrative paradigm is called multiscale/multiphysics modelling and simulation (MMS). It is a general paradigm that applies to many domains in physical sciences involving modelling and simulation, such as engineering, material science, physics, astronomy and environment [2].

Objective. The purpose of this study is to answer the following questions:

1. Which MMS software frameworks exist in the literature and market?
2. What main software engineering mechanisms do they employ to allow integration/extension/coupling, expressed briefly?

Related work. There are an overwhelmingly large number of MMS software frameworks in the literature and market, whether developed by small academic groups, open communities or commercial institutions. Several listings of these frameworks are presented within the individual papers of the frameworks as related work [3, 4, 5], and in a couple of survey papers [2, 6, 7, 8]. To our best knowledge, there has not been a comprehensive bibliography of these tools in the literature.

2 Method and Results

In this section, we briefly explain our method and results. Our method in principle combines Systematic Literature Review (SLR) [9], which involves a database search as a first step, and snowballing [10], which involves inspecting the selected papers' references and external secondary sources as a second step.

Search process Following the SLR method, we used a bibliographic search as the first step. We preferred the Scopus database because of its wide coverage of items in various scientific areas and powerful search interface. In general we aimed to reduce the number of articles to consider, while maintaining a good set of articles as the starting point. We used the Advanced Search facility of Scopus to identify potential articles that:

1. have explicitly *'multiphysics', 'multi-physics', 'multiscale'* or *'multi-scale'* in their title, abstract or keywords,
2. have explicitly *'modeling', 'modelling'* or *'simulation'* in their title, abstract or keywords,
3. have *'framework', 'platform', 'environment', 'toolkit', 'integration', 'coupling'* or *'coupler'* in their title, abstract or keywords,
4. are published from *2000 onwards*, and
5. are in the subject area of *physical sciences* as defined by Scopus.

This search, performed on 16.02.2015, resulted in 6000+ items. As a next step to further narrow down the scope, we filtered out the articles with less than 0.5 citations per year. We treated the publications of the last two years (2013 onwards) differently; we included them all regardless of their citation count. With this filter we obtained 4000+ items, which arguably contain (relatively) more prominent articles for our consideration.

Selection criteria. We did a cross-reading of the titles and abstracts of the selected items to identify a second set of better candidates to examine closer. This activity in turn resulted in 400+ items. We performed a detailed examination, to ensure that article presents software that:

1. is a software framework, rather than a conceptual or mathematical framework,
2. explicitly involves modelling and simulation, and
3. explicitly promotes features of a framework, e.g. extension/extendability, integration, interoperability, coupling, modularity, genericness or reuse.

We deliberately excluded the software that e.g. fulfil only two out of three. Among the excluded are commercial tools, which most typically are closed source and monolithic. The first SLR part of our work resulted in 100+ items to be included in our bibliography.

Snowballing. For each selected item, we consulted the main article of interest, plus additional resources such as the software website, repository, secondary publications, etc. Finally we checked the references in the related work sections of these articles, and consulted a few additional resources (literature, websites, etc.), to further identify relevant frameworks for our bibliography. As a result of this snowballing activity, we could identify an additional 40+ items, which we added in the bibliography.

Final results and annotations. Eventually, we collected 140+ items, namely MMS software frameworks and important related technologies such as domain specific languages and ontologies. We examined all the relevant material (publications, websites, etc.) for each individual framework, to identify the main mechanisms used for software integration/extension/ coupling. This information was documented in the extensive bibliography in the form of annotations below each reference.

The full list of references at the end of this paper are organized according to the subject area categorization provided by Scopus where possible, or assigned by us after manual inspection. The final categories we chose are: engineering (also including computer science/software engineering/mathematics/multidisciplinary, because those areas tend to generic computational frameworks applicable to many areas of engineering as well), physics and astronomy, earth and environmental sciences (merging earth and planetary sciences with engineering), energy, chemistry/chemical engineering, material science, and life/health sciences. Note that although we deliberately excluded life/health sciences from our bibliographic search (i.e. we restricted our search to physical sciences only), we found many of those listed in other domains as well, because of their multidisciplinary nature.

Threats to Validity The elements of our literature survey has some natural threats to validity: restricting the literature search to Scopus database only, the limited selection of search keywords and the elimination of the least cited group of articles might all lead to the possible exclusion of MMS frameworks. This shortcoming is partly remedied by the snowballing where we track the references and external sources to identify more frameworks. Furthermore, the huge amount of manual work done might include subjectivity or erroneous judgement. There is no feasible automated solution for this, but this study could be further supported by empirical studies such as surveys and questionnaires to increase the coverage and quality.

3 Conclusion

In this paper we report an extensive survey that we conducted on MMS software frameworks. After a rigorous search, we discovered a significantly large number of those, and listed them in an annotated bibliography. We annotated the frameworks regarding their main mechanisms for software integration/extension/coupling. Detailed analysis and discussion of the results is out of scope for this paper, and left as future work.

Acknowledgements

The research leading to these results has been funded by EU programme FP7-NMP-2013-SMALL-7 under grant agreement number 604279 (MMP).

References

[1] D. L. Brown, J. Bell, D. Estep, W. Gropp, B. Hendrickson, S. Keller-McNulty, D. Keyes, J. T. Oden, L. Petzold, and M. Wright. *Applied Mathematics at the U.S. Department of Energy: Past, Present and a View to the Future*. Feb 2008.

[2] D. Groen, S. J. Zasada, and P. V. Coveney. Taxonomy of Multiscale Computing Communities. *2011 IEEE Seventh International Conference on e-Science Workshops*, pages 120–127, December 2011.

[3] B. Patzák, D. Rypl, and J. Kruis. MuPIF - A distributed multi-physics integration tool. *Advances in Engineering Software*, 60-61:89–97, June 2013.

[4] E. De Sturler and J. Hoeflinger. A new approach to software integration frameworks for multi-physics simulation codes *. *The Architecture of Scientific Software*, pages 87–104, 2001.

[5] S. F. Portegies Zwart et al. Multi-physics simulations using a hierarchical interchangeable software interface. *Computer Physics Communications*, 184(3):456–468, March 2013.

[6] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, et al. Multiphysics simulations challenges and opportunities. *International Journal of High Performance Computing Applications*, 27(1):4–83, 2013.

[7] J. G. Michopoulos, C. Farhat, and J. Fish. Modeling and Simulation of Multiphysics Systems. *Journal of Computing and Information Science in Engineering*, 5(3):198, 2005.

[8] A. Yang and W. Marquardt. An ontological conceptualization of multiscale models. *Computers & Chemical Engineering*, 33(4):822–837, April 2009.

[9] S. Keele. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, EBSE Technical Report EBSE-2007-01, 2007.

[10] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, page 38. ACM, 2014.

Frameworks in Engineering, Mathematics and Computer Science

[11] J.J. Alonso, S. Hahn, F. Ham, M. Herrmann, G. Iaccarino, G. Kalitzin, P. LeGresley, K. Mattsson, G. Medic, P. Moin, H. Pitsch, J. Schlüter, M. Svärd, E. Van Der Weide, D. You, and X. Wu. Chimps: A high-performance scalable module for multi-physics simulations. *Collection of Technical Papers - AIAA/ASME/SAE/ASEE 42nd Joint Propulsion Conference*, 11:9091–9118, 2006. cited By 17.

Presents a Coupler for High-Performance Integrated Multi-Physics Simulations library (CHIMPS), a code coupling environment to facilitate the setup and execution of large-scale integrated simulations, from Stanford University. It relies on a Fortran or Python driver script that orchestrates codes through a common API.

[12] E.L. Blades, R.S. Miskovish, E.A. Luke, E.M. Collins, and A.G. Kurkchubashe. A multiphysics simulation capability using the simulia co-simulation engine. *20th AIAA Computational Fluid Dynamics Conference 2011*, 2011. cited By 3.

Presents an integration approach using SIMULIA Co-Simulation Engine (CSE), a software framework that allows the coupling of multiple simulation domains by coupling solvers in a synchronized manner. CSE allows a flexible C interface to bind C, C++ and Fortran codes, and possesses a runtime environment to orchestrate/synchronize the co-simulation.

[13] M. Bogdanova, S. Belousov, I. Valuev, A. Zakirov, M. Okun, D. Shirabaykin, V. Chorkov, P. Tokar, A. Knizhnik, B. Potapkin, A. Bagaturyants, K. Komarova, M.N. Strikhanov, A.A. Tishchenko, V.R. Nikitenko, V.M. Sukharev, N.A. Sannikova, and I.V. Morozov. Simulation platform for multiscale and multiphysics modeling of oleds. *Procedia Computer Science*, 29:740–753, 2014.

Presents an integrated framework for multiscale and multiphysics modeling of Organic Light Emitting Diodes (OLEDs), from Kintech Lab. It adopts a data-centric approach with data port I/O specification for modules to be used in the workflow editor and data conversion with JavaScript.

[14] N. Bombieri, D. Drogoudis, G. Gangemi, R. Gillon, E. Macii, M. Poncino, S. Rinaudo, F. Stefanni, D. Trachanis, and M. Van Helvoort. Smac: Smart systems co-design. *Proceedings - 16th Euromicro Conference on Digital System Design, DSD 2013*, pages 253–259, 2013. cited By 1.

Presents SMAC (SMArt systems Co-design), a flexible platform for multiscale electronics device modeling. It allows modeling subsystems in different levels (e.g. using VHDL, Verilog-A, Matlab) that are linked to their C++ implementations, which can be abstracted/refined/integrated.

[15] J. Borgdorff, M. Mamonski, B. Bosak, K. Kurowski, M. Ben Belgacem, B. Chopard, D. Groen, P.V. Coveney, and A.G. Hoekstra. Distributed multiscale computing with muscle 2, the multiscale coupling library and environment. *Journal of Computational Science*, 5(5):719–731, 2014. cited By 4.

Presents the Multiscale Coupling Library and Environment (MUSCLE 2), a portable framework for multiscale modeling and simulation, developed within the EU MAPPER project. It allows the integration of Java, C/C++, Python, Matlab and Scala code in a component-based architecture. Configuration of the coupling is done in Ruby based on CxA formalism via various mechanisms such as MPI.

[16] J. Brown, M.G. Knepley, D.A. May, L.C. McInnes, and B. Smith. Composable linear solvers for multiphysics. *Proceedings - 2012 11th International Symposium on Parallel and Distributed Computing, ISPDC 2012*, pages 55–62, 2012. cited By 4.

Presents the multiphysics capabilities of PETSc (The Portable, Extensible Toolkit for Scientific computing). PetSc has a flexible architecture, with interfaces with C, C++, Fortran 77/90, Python, and MATLAB codes and MPI-based communication.

[17] D. Brutzman, M. Zyda, J. M. Pullen, K. L. Morse, S. Fouskarinis, D. Drake, D. Moen, C. Blais, A. Kapolka, and D. McGregor. Extensible modeling and simulation framework (xmsf) challenges for web-based modeling and simulation. *Findings and Recommendations Report: Technical Challenges Workshop, Strategic Oppurtunities Symposium*, 2002.

Presents XMSF (Extensible Modeling and Simulation Framework), a service-oriented framework for multi-application simulations. It promotes Web Services and XML, and complies with several standards such as HLA.

[18] T. Bulatewicz and J. Cuny. A domain-specific language for model coupling. *Proceedings - Winter Simulation Conference*, pages 1091–1100, 2006. cited By 5.

Presents the Coupling Description Language, a visual domain-specific language for model coupling, from University of Oregon. It allows the description of model interactions based on high level Potential Coupling Interfaces, and interoperability of models independent of the source code.

[19] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski. Toward a common component architecture for high-performance scientific computing. *IEEE International Symposium on High Performance Distributed Computing, Proceedings*, pages 115–124, 1999. cited By 49.

Presents the Common Component Architecture (CCA), a component model for interoperability among scientific software. It can be considered as a domain-specific variant of CORBA for scientific computing. Several concrete implementations include CCaffeine and XCAT.

[20] D. De Cecchis, L.A. Drummond, and J.E. Castillo. Design of a distributed coupling toolkit for high performance computing environment. *Mathematical and Computer Modelling*, 57(9-10):2267–2278, 2013. cited By 1.

Presents the Distributed Coupling Toolkit (DCT), a coupling framework for high performance computing simulations. It provides a distributed communication protocol based on the producer/consumer model and data transformations for interoperability.

[21] E. de Sturler, J. Hoeflinger, L. Kale, and M. Bhandarkar. A new approach to software integration frameworks for multi-physics simulation codes. *IFIP Advances in Information and Communication Technology*, 60:87–102, 2001.

Presents a code integration framework for multiphysics simulation, from University of Illinois at Urbana-Champaign. It proposes a system built on top of Charm++ framework as the communication/load balancing infrastructure, and annotating code with Code Description and Orchestration Description for (possibly automatic) translation into the framework.

[22] C.C. Douglas and Y. Efendiev. A dynamic data-driven application simulation framework for contaminant transport problems. *Computers and Mathematics with Applications*, 51(11):1633–1646, 2006. cited By 8.

Presents a dynamic data-driven application simulation framework (DDDAS) for multiscale transport problems. It adopts a data-centric approach, providing a distributed data management system to integrate
.

[23] T. Dupont, J. Hoffman, C. Johnson, R. C. Kirby, M. G. Larson, A. Logg, and L. R. Scott. *The FEniCS project.* Chalmers Finite Element Centre, Chalmers University of Technology, 2003.

Presents FEniCS project, a free software project for the automation of computational mathematical modeling. It contains a mathematical description language (Unified Form Language) for computational modeling, and compiles it into optimal C++ code via its Dolfin module, interoperable with a number of external software.

[24] J.-L. Falcone, B. Chopard, and A. Hoekstra. Mml: Towards a multiscale modeling language, 2010. cited By 21.

Presents Multiscale Modeling Language (MML) i.e. a description language aiming at specifying the architecture of a multiscale simulation program. It allows a graphical or XML-based textual representation of multiscale models.

[25] C. Feichtinger, S. Donath, H. Köstler, J. Götz, and U. Rüde. Walberla: Hpc software design for computational engineering simulations. *Journal of Computational Science*, 2(2):105–112, 2011. cited By 21.

Presents WALBERLA (widely applicable Lattice-Boltzmann from Erlangen), an object-oriented framework for multiphysics simulations, from Friedrich-Alexander-Universität Erlangen-Nürnberg.Written in C++, it relies on a modular architecture with an extendible class hierarchy, and MPI-based communication for integration of simulation codes.

[26] T. Fischbacher, M. Franchin, G. Bordignon, and H. Fangohr. A systematic approach to multiphysics extensions of finite-element-based micromagnetic simulations: Nmag. *IEEE Transactions on Magnetics*, 43(6):2896–2898, 2007. cited By 84.

Presents Nmag, a flexible finite element micromagnetic simulation package developed at University of Southampton. It is written in Python and connects natively with Nsim simulation library, which in turn interacts with external tools (i.e. PETSc) through an Objective CAML interface.

[27] L. Fornasier, H. Rieger, U. Tremel, and E. van der Weide. Time-dependent aeroelastic simulation of rapid manoeuvring aircraft. *40th AIAA Aerospace Sciences Meeting and Exhibit*, 2002. cited By 5.

Presents JULIUS SimServer, an integration platform for multiphysics aeroelastic simulations, from Stanford University. Written in C++, it has a modular object-oriented architecture with MPI-based communication.

[28] B. FrantzDale, S.J. Plimpton, and M.S. Shephard. Software components for parallel multiscale simulation: An example with lammps. *Engineering with Computers*, 26(2):205–211, 2010. cited By 10.

Presents an extension of the large-scale atomistic/molecular massively parallel simulator (LAMMPS), an atomistic simulation software, from

Sandia National Laboratories. Written in C++, LAMMPS and its extension allows integration with other codes via its (hereby extended) library interface and a Python wrapper.

[29] B. Gatzhammer, M. Mehl, and T. Neckel. A coupling environment for partitioned multiphysics simulations applied to fluid-structure interaction scenarios. *Procedia Computer Science*, 1(1):681–689, 2010. cited By 8.

Presents preCICE(Precise Code Interaction Coupling Environment), a open-source general-purpose multiphysics coupling tool, from Technische Universitaet Muenchen. It relies on black-box wrapping of existing tools, data mapping and MPI-based communication for interoperability.

[30] W.D. Henshaw. Overture: An object-oriented framework for overlapping grid applications. *32nd AIAA Fluid Dynamics Conference and Exhibit*, 2002. cited By 2.

Presents Overture, an object-oriented environment for solving multigrid partial differential equations, from Los Alamos National Laboratories. Written in C++, it relies on a rich class hierarchy and overloaded operators for extension by non-programmers.

[31] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, and K.S. Stanley. An overview of the trilinos project. *ACM Transactions on Mathematical Software*, 31(3):397–423, 2005. cited By 252.

Presents Trilinos, an open source object-oriented framework for the solution of large-scale, complex multiphysics engineering and scientific problems, developed by Sandia National Laboratory and a large open community. It promotes the interoperability between existing software and uses software engineering practices for the development of new scientific packages; maintaining modularity on the level of individual packages written in C++.

[32] B. Müller, K.L. Morse, M. Lightner, R. Little, and R. Lutz. Hla evolved - a summary of major technical improvements, 2008.

Presents High Level Architecture (HLA) Evolved, an extension to the IEEE 1516 standard for distributed simulations. In addition to the original features of object models and run-time infrastructure middleware, HLA Evolved adds XML/XML Schema support, Web Services support, enhanced modularity and many more.

[33] A.G. Hoekstra, E. Lorenz, J.-L. Falcone, and B. Chopard. Toward a complex automata formalism for multiscale modeling. *International Journal for Multiscale Computational Engineering*, 5(6):491–502, 2007. cited By 34.

Presents CxA (Complex Automata), a formalism for modeling complex multiscale systems, from University of Amsterdam.

[34] R.D. Hornung and S.R. Kohn. Managing application complexity in the samrai object-oriented framework. *Concurrency Computation Practice and Experience*, 14(5):347–368, 2002. cited By 85.

Presents Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI), an object-oriented framework for parallel structured adaptive mesh refinement for large-scale multiphysics problems, from Lawrence Livermore National Laboratory. It is written in C++ and facilitates object-oriented design patterns to achieve flexibility and extendibility in terms of new applications.

[35] E.N Houstis, A.C. Catlin, P. Tsompanopoulou, D. Gottfried, G. Balakrishnan, K. Su, and J.R. Rice. Gasturbnlab: A multidisciplinary problem solving environment for gas turbine engine design on a network of nonhomogeneous machines. *Journal of Computational and Applied Mathematics*, 149(1):83–100, 2002. cited By 14.

Presents GasTurbnLab, a multidisciplinary problem solving environment for gas turbine engine design, from Purdue University. It has an agent-based architecture, where legacy code in Fortran,C can be wrapped as Java objects with well defined interfaces, collaborating via the agent middleware Grasshopper.

[36] H. Jasak. Openfoam: Open source cfd in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2):89–94, 2009. cited By 23.

Presents OpenFOAM, an open-source object-oriented framework for computational fluid dynamics, from OpenFOAM Foundation. Written in C++, it has a modular and extendible object-oriented architecture, easy for integrating new code into the framework. It has also features like dynamic linking and symbolic description of equations to promote flexibility. It supports a wide range of (commercial) data formats via built-in converters.

[37] X. Jiao, M.T. Campbell, and M.T. Heath. Roccom: An object-oriented, data-centric software integration framework for multiphysics simulations. *Proceedings of the International Conference on Supercomputing*, pages 358–367, 2003. cited By 11.

Presents Roccom, an object-oriented software integration framework for rocket simulation, from University of Illinois at Urbana-Champaign. Written in C++, it has an object-oriented architecture, and emulates some component-based mechanisms such as interface definition and run-time bindings to support interoperability with Fortran90 and C++ simulation code. It is a central part of the Illinois Rocstar suite.

[38] W. Joppich and M. Kürschner. Mpcci - a tool for the simulation of coupled applications. *Concurrency Computation Practice and Experience*, 18(2):183–192, 2006. cited By 27.

Presents MpCCI (mesh-based parallel code coupling interface), a commercial framework for the coupling of different simulation codes, from Fraunhofer SCAI. Written in C++, it allows the integration of C++ simulation code natively, and C/Fortran code through an interface; using MPI to connect the codes orchestrated by a central coupling server. MpCCI further provides interfaces to with many popular simulation tools such as Abaqus, Matlab and OpenFOAM.

[39] A. Lani, T. Quintino, D. Kimpe, H. Deconinck, S. Vandewalle, and S. Poedts. The coolfluid framework: Design solutions for high performance object oriented scientific computing software. *Lecture Notes in Computer Science*, 3514(I):279–286, 2005.

Presents CoolFluid (Computational Object-Oriented Library for Fluid Dynamics), an open object-oriented framework for multi-physics simulations, from von Karman Institute. Written in C++ with a Python interface, it has a plug-in system where objects are self-registered and self-configured, i.e. a component-based architecture, and applies many design patterns to promote extendibility. It further includes a domain specific embedded language for describing finite element simulations.

[40] W. Li, A. Monti, M. Luo, and R.A. Dougal. Vpnet: A co-simulation framework for analyzing communication channel effects on power systems. *2011 IEEE Electric Ship Technologies Symposium, ESTS 2011*, pages 143–149, 2011. cited By 18.

Presents VPNET, a co-simulation framework for power systems and communication networks, from RWTH Aachen and University of South Carolina. It couples the Virtual Test Bed framework via its COM interface to OPNET simulator via inter-process interface IPC).

[41] H.G. Matthies, R. Niekamp, M. Hautefeuille, D. Jürgens, C. Kassiotis, T. Srisupattarawanit, J.-B. Colliat, and A. Ibrahimbegović. Elements of a computational framework for coupled simulations. *GAMM Mitteilungen*, 33(1):73–78, 2010. cited By 2.

Presents a component-based approach for coupling multiscale simulations, from Technische Universität Braunschweig. It has a component-based architecture based on Component Template Library, which provides the standard component model and communication infrastructure.

[42] J. Michopoulos, P. Tsompanopoulou, E. Houstis, C. Farhat, M. Lesoinne, J. Rice, and A. Joshi. On a data-driven environment for multiphysics applications. *Future Generation Computer Systems*, 21(6):953–968, 2005. cited By 21.

Presents a data-driven environment for multiphysics applications (DDEMA) as a multidisciplinary problem solving environment, from Purdue University and other institutions. It aims to develop an agent-based system as middleware to integrate legacy codes and real time data sources (e.g. sensors).

[43] M. Mita, S. Maruyama, Y. Yi, K. Takahashi, H. Fujita, and H. Toshiyoshi. Multiphysics analysis for micro electromechanical systems based on electrical circuit simulator. *IEEJ Transactions on Electrical and Electronic Engineering*, 6(2):180–189, 2011. cited By 17.

Presents a multiphysics simulation of microelectromechanical systems (MEMS) on the open source framework Qucs (Quite Universal Circuit Simulator). It proposes a model-based approach on hierarchical modeling in Verilog-A, and code generation in C++ for integration.

[44] J.H. Nie, D.A. Hopkins, Y.T. Chen, and H.T. Hsieh. Development of an object-oriented finite element program with adaptive mesh refinement for multi-physics applications. *Advances in Engineering Software*, 41(4):569–579, 2010. cited By 5.

Presents an object-oriented framework for numerical analysis of multi-physics applications, from US Army Research Laboratory. Written in Fortran 2003, it relies on a modular object-oriented hierarchy for flexibility.

[45] J.D. Oldham. Pooma: A c++ toolkit for high-performance parallel scientific computing. *National Aeronautics and Space Administration Publication*, 2002.

Presents POOMA (Parallel Object-Oriented Methods and Applications), a multidomain framework for parallel simulations, from U.S. Department of Energy. Written in C++, it relies on a modular layered class hierarchy and object-oriented mechanisms for flexibility and extension.

[46] F. Palacios, M.R. Colonno, A.C. Aranake, A. Campos, S.R. Copeland, T.D. Economon, A.K. Lonkar, T.W. Lukaczyk, T.W.R. Taylor, and J.J. Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. *51st AIAA Aerospace Sciences*

Meeting including the New Horizons Forum and Aerospace Exposition 2013, 2013. cited By 1.

Presents Stanford University Unstructured (SU^2), an open-source integrated computational environment for multi-physics simulation and design. Written in C++, it has a flexible object-oriented architecture and Python conversion scripts for integration and extension.

[47] B. Patzák and Z. Bittnar. Design of object oriented finite element code. *Advances in Engineering Software*, 32(10-11):759–767, 2001. cited By 106.

Presents OOFEM (Object-Oriented Finite Element Method), a multiphysics finite element environment, from Czech Technical University. Written in C++, it has an extendible modular class hierarchy, which establishes a basis for integration of new codes through extension.

[48] B. Patzák, D. Rypl, and J. Kruis. Mupif-a distributed multi-physics integration tool. *Advances in Engineering Software*, 60-61:89–97, 2013. cited By 1.

Presents the Multi-Physics Integration Framework (MuPIF), an opensource object-oriented framework for integrating independently developed multi-physics simulations. Written in Python, it has a flexible class hierarchy plus the scripting capabilities of Python; and uses SWIG/Boost wrappers to incorporate code written in C/C++.

[49] S. Picault and P. Mathieu. An interaction-oriented model for multi-scale simulation. *IJCAI International Joint Conference on Artificial Intelligence*, pages 332–337, 2011. cited By 6.

Presents an environment for agent-based multiscale simulations, from University Lille 1. It proposes a formalism, PADAWAN, for agent-based modelling and its integration to the interaction-based (IODA) simulation platform, JEDI.

[50] G. Quesnel, R. Duboz, and E. Ramat. The virtual laboratory environment - an operational framework for multi-modelling, simulation and analysis of complex dynamical systems. *Simulation Modelling Practice and Theory*, 17(4):641–653, 2009. cited By 49.

Presents the Virtual Laboratory Environment (VLE), a framework for multi-modelling and simulation of complex dynamical systems. Written in C++, it adopts on a layered agent-based architecture supporting various formalisms and compatibility of other languages such as Java, Fortran and Python.

[51] A.C. Robinson, T.A. Brunner, S. Carroll, Richarddrake, C.J. Garasi, T. Gardiner, T. Haill, H. Hanshaw, D. Hensinger, D. Labreche, R. Lemke, E. Love, C. Luchini,

S. Mosso, J. Niederhaus, C.C. Ober, S. Petney, W.J. Rider, G. Scovazzi, O.E. Strack, R. Summers, T. Trucano, V.G. Weirs, M. Wong, and T. Voth. Alegra: An arbitrary lagrangian-eulerian multimaterial, multiphysics code. *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008. cited By 1.

Presents ALEGRA, an arbitrary Lagrangian-Eulerian multimaterial, multiphysics coupling framework from Sandia National Laboratories. Written in C++, it relies on object-orientation and templates for flexibility and reuse.

[52] K. Rycerz, M. Bubak, and P.M.A. Sloot. Dynamic interactions in hla component model for multiscale simulations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5102 LNCS(PART 2):217–226, 2008. cited By 1.

Presents CompoHLA, an implementation of HLA component model for distributed multiscale simulations, from . It applies the distributed federation component model of HLA on Grid, to achieve interoperability and reusability.

[53] R. Schmidt, N. Belcourt, R. Hooper, and R. Pawlowski. An introduction to lime 1.0 and its use in coupling codes for multiphysics simulations. *SAND2011-8524, Sandia National Laboratories, Albuquerque, New Mexico*, 2011.

Presents LIME(Lightweight Integrating Multiphysics Environment for coupling codes), a small coupling framework for multiphysics simulation, from Sandia National Laboratories. Written in C++, it uses object-oriented mechanisms and XML-based configuration to specify couplings. It also supports bindings from Fortran, and interfaces with simulation tools such as Trilinos.

[54] JamesR. Stewart and H.Carter Edwards. The sierra framework for developing advanced parallel mechanics applications. In L. T. Biegler, M. Heinkenschloss, O. Ghattas, and B. van Bloemen Waanders, editors, *Large-Scale PDE-Constrained Optimization*, volume 30 of *Lecture Notes in Computational Science and Engineering*, pages 301–315. Springer Berlin Heidelberg, 2003.

Presents SIERRA, an object-oriented computational framework for developing complex multiphysics applications, from Sandia National Laboratories. Written in C++, it has an object-oriented layered architecture and relies on object-oriented techniques for extension.

[55] M. Storti, N. Nigro, R. Paz, L. Dalcin, and E. Lopez. Petsc-fem, a general purpose, parallel, multi-physics fem program. *International Center of Computational Method in Engineering (CIMEC), Argentina http://www. cimec. org. ar/petscfem*, 2007.

Presents PETSc-FEM, a general purpose multi-physics FEM program for computational fluid dynamics based on PETSc, from CIMEC, Argentina. Written mainly in C++ and Perl, it relies object-oriented mechanisms and predefined hooks (a concept borrowed from Emacs) to dynamically load extensions at run-time.

[56] H. Talebi, M. Silani, S.P.A. Bordas, P. Kerfriden, and T. Rabczuk. A computational library for multiscale modeling of material failure. *Computational Mechanics*, 53(5):1047–1071, 2014. cited By 8.

Presents PERMIX, an object-oriented framework for multiscale modeling of material failure, from Bauhaus-Universitaet Weimar. Written in Fortran 2003, it uses object-oriented mechanisms for extension and bindings to C/C++ for interfacing with various other tools such as LAMMPS.

[57] C. Wang. Insights from developing a multidisciplinary design and analysis environment. *Computers in Industry*, 65(4):786–795, 2014.

Presents the multidisciplinary design and analysis (MDA) environment, which allows the integration of tools for complex engineering tasks, from Northeastern University China. It follows a component-based approach: it wraps legacy applications using surrogate objects in a component model, maintains them in a dynamic registry, and uses data/model mappings and COM/DCOM communication infrastructure for interoperability.

[58] A. Xiao, K.M. Bryden, and D.S. McCorkle. Ve-suite: A software framework for design-analysis integration during product realization. *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005*, 3 B:859–867, 2005.

Presents VE-Suite, an open source virtual engineering software toolkit, from Iowa State University. Written in C++, it uses component-based integration via CORBA, and an ontology-based description of the objects for semantical integration.

[59] F. Zhang, C. Docan, H. Bui, M. Parashar, and S. Klasky. Xpressspace: A programming framework for coupling partitioned global address space simulation codes. *Concurrency Computation Practice and Experience*, 26(3):644–661, 2014.

Presents XpressSpace, a data-centric programming framework for coupling multiphysics simulation codes, from Rutgers University. It extends the PGAS programming framework, with specific features for multiphysics simulations such as distributed and parallel data management and sharing.

Frameworks in Physics and Astronomy

[60] W.R. Elwasif, D.E. Bernholdt, A.G. Shet, S.S. Foley, R. Bramley, D.B. Batchelor, and L.A. Berry. The design and implementation of the swim integrated plasma simulator. *Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing, PDP 2010*, pages 419–427, 2010. cited By 6.

> Presents The Integrated Plasma Simulator (IPS), a component-based framework for plasma simulations, within SWIM project. Written in Python, it provides framework services and RPC-based communication of components inside wrappers. It further revolves around a data-centric approach where the plasma state files (in NetCDF format) are shared among many components.

[61] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf. The cactus framework and toolkit: Design and applications. In *High Performance Computing for Computational Science—VECPAR 2002*, pages 197–227. Springer, 2003.

> Presents Cactus, an open-source multiphysics framework for astrophysics, from Max Planck Institute for Gravitational Physics. It is written in C and emphasizes its highly modular and extendable component-based architecture to support the integration of legacy codes, in Fortran and C++ as well. It has an advanced declarative specification language (Cactus Configuration Language) for declaring the extension points and properties for compliance and interoperability among other extensions.

[62] J.D. Halverson, T. Brandes, O. Lenz, A. Arnold, S. Bevc, V. Starchenko, K. Kremer, T. Stuehn, and D. Reith. Espresso++: A modern multiscale simulation package for soft matter systems. *Computer Physics Communications*, 184(4):1129–1149, 2013. cited By 12.

> Presents Extensible Simulation Package for Research on Soft matter systems (ESPResSo++), is an open-source object-oriented simulation package for the simulation of condensed soft matter systems, from Max Planck Institute for Polymer Research. Its kernel is implemented in C++ and it has a Python interface operating with Boost wrapper; and it aims at high extendibility with scripting and inheritance.

[63] C. Nieter and J.R. Cary. Vorpal: A versatile plasma simulation code. *Journal of Computational Physics*, 196(2):448–473, 2004. cited By 328.

> Presents VORPAL, an object-oriented framework for plasma simulation, from University of Colorado. Written in C++, it uses meta-template programming and Python interface for flexibility and extension. It has later been commercialized and renamed as VSim.

[64] Y.A. Omelchenko and H. Karimabadi. Hypers: A unidimensional asynchronous framework for multiscale hybrid simulations. *Journal of Computational Physics*, 231(4):1766–1780, 2012. cited By 6.

> Presents HYPERS (HYbrid Particle Event-Resolved Simulator), an event-driven multiscale hybrid simulation framework for plasma, from SciberQuest. It is written in C++/Fortran77, and integrates discrete-event simulation codes in an event handling and scheduling infrastructure.

[65] S.F. Portegies Zwart, S.L.W. McMillan, A. Van Elteren, F.I. Pelupessy, and N. De Vries. Multi-physics simulations using a hierarchical interchangeable software interface. *Computer Physics Communications*, 184(3):456–468, 2013. cited By 14.

> Presents MUSE (MUlti-physics Software Environment) and its implementation AMUSE, an open-source astrophysics framework for the integration of multiphysics simulations, from Leiden Observatory. It is written in Python and supports the integration of C++ and Fortran legacy code by implementing object-oriented wrapper interfaces around them, and orchestration is handled in the Python code.

[66] T.D. Scheibe, A.M. Tartakovsky, D.M. Tartakovsky, G.D. Redden, and P. Meakin. Hybrid numerical methods for multiscale simulations of subsurface biogeochemical processes. *Journal of Physics: Conference Series*, 78(1), 2007. cited By 8.

> Presents a multiscale modeling framework for flow and transport problems that integrates models for physics and chemistry. It uses the component standard CCA for interoperability.

[67] J. Torras, Y. He, C. Cao, K. Muralidharan, E. Deumens, H.-P. Cheng, and S.B. Trickey. Pupil: A systematic approach to software integration in multi-scale simulations. *Computer Physics Communications*, 177(3):265–279, 2007. cited By 12.

> Presents Program for User Package Interfacing and Linking (PUPIL), a multi-scale simulation framework to integrate existing software, from University of Florida. Written in Java, it is an object-oriented system that uses wrappers (notably Java Native Interface for C) around the legacy code and CORBA communication among them. It also stores the simulation directives in an XML file for interoperability among the various modules.

[68] M.J. Turk, B.D. Smith, J.S. Oishi, S. Skory, S.W. Skillman, T. Abel, and M.L. Norman. Yt: A multi-code analysis toolkit for astrophysical simulation data. *Astrophysical Journal, Supplement Series*, 192(1), 2011. cited By 70.

Presents yt, an open source astrophysical analysis and visualization toolkit, developed by a community including University of California. It is written in Python and allows data-oriented interoperability with other simulation tools such as FLASH.

[69] J. Van Dijk, K. Peerenboom, M. Jimenez, D. Mihailova, and J. Van Der Mullen. The plasma modelling toolkit plasimo. *Journal of Physics D: Applied Physics*, 42(19), 2009. cited By 30.

Presents Plasimo, a multiphysics plasma modeling toolkit from Eindhoven University of Technology. It is object-oriented, implemented in C++, and has recently incorporated XML-based technologies to describe data and web services to introduce better interfaces to the toolkit.

Frameworks in Energy

[70] I. Agapov, G. Geloni, S. Tomin, and I. Zagorodnov. Ocelot: A software framework for synchrotron light source and fel studies. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 768:151–156, 2014.

Presents OCELOT, a multiphysics toolkit for accelarator simulations, from European XFEL. It is written in Python and relies on its high-level scripting capabilities for the integration of data and functionality. It potentially proposes interfacing with other models such as Accelerator Markup Language (AML).

[71] J.R. Cary, J. Candy, J. Cobb, R.H. Cohen, T. Epperly, D.J. Estep, S. Krasheninnikov, A.D. Malony, D.C. McCune, L. McInnes, A. Pankin, S. Balay, J.A. Carlsson, M.R. Fahey, R.J. Groebner, A.H. Hakim, S.E. Kruger, M. Miah, A. Pletzer, S. Shasharina, S. Vadlamani, T.D. Wade-Stein, D. Rognlien, A. Morris, S. Shende, G.W. Hammett, K. Indireshkumar, A. Yu Pigarov, and H. Zhang. Concurrent, parallel, multiphysics coupling in the facets project. *Journal of Physics: Conference Series*, 180(1), 2009. cited By 6.

Presents the Framework Application for Core-Edge Transport Simulations (FACETS), a project aiming to develop a multiphysics coupling framework for fusion simulations. Written in C++, it uses a component hierarchy with well defined interfaces, wrap legacy code (e.g. using Forthon) and integrate into the framework.

[72] J. Cummings, J. Lofstead, K. Schwan, A. Sim, A. Shoshani, C. Docan, M. Parashar, S. Klasky, N. Podhorszki, and R. Barreto. Effis: An end-to-end framework for fusion integrated simulation. *Proceedings of the 18th Euromicro*

Conference on Parallel, Distributed and Network-Based Processing, PDP 2010, pages 428–434, 2010. cited By 3.

Presents End-to-end Framework for Fusion Integrated Simulation (EFFIS), a multiphysics framework for plasma simulations. It relies on an adaptable I/O system that takes over the individual I/O tasks in the integrated codes into a common XML-based mechanism; thus coupling the codes in a data-centric manner.

[73] R.A. Dougal, S. Liu, L. Gao, and M. Blackwelder. Virtual test bed for advanced power sources. *Journal of Power Sources*, 110(2):285–294, 2002. cited By 11.

Presents the Virtual Test Bed (VTB), a multidisciplinary open framework for the simulation and virtual prototyping of advanced power systems, developed by University of South Carolina. It is written in .NET framework, and allows integration of models and services via XML-based specifications and COM/.NET interfaces. Recent developments include a UNICORE-based distributed version.

[74] A. Dubey, K. Antypas, M.K. Ganapathy, L.B. Reid, K. Riley, D. Sheeler, A. Siegel, and K. Weide. Extensible component-based architecture for flash, a massively parallel, multiphysics simulation code. *Parallel Computing*, 35(10-11):512–522, 2009. cited By 66.

Presents FLASH, an open source multiscale multiphysics simulation code originally for the energy domain from University of Chicago. Written in Fortran90, it has an extensible component-based (not exactly as in Component-Based Software Engineering terminology, but rather to mean modules) architecture that allows the integration of legacy code.

[75] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009. cited By 69.

Presents Multiphysics Object Oriented Simulation Environment (MOOSE), an open source parallel computational framework for nuclear simulations, from Idaho National Laboratory. It is organized as C++ modules and classes. It has further domain-specific extensions called MARMOT, BISON, PRONGHORN and PyMOOSE.

[76] O. Meneghini and L. Lao. Integrated modeling of tokamak experiments with omfit. *Plasma and Fusion Research*, 8(SPL.ISS.1), 2013. cited By 1.

Presents One Modeling Framework for Integrated Tasks (OMFIT), a framework for data-centric integration of fusion simulations, from Oak Ridge Associated Universities. Written in Fortran, it proposes a central hierarchical data structure keeping track of files, data and script; and allows shared access by different simulation codes. Workflow is specified with Python scripts.

[77] A. Siegel, T. Tautges, A. Caceres, D. Kaushik, P. Fischer, G. Palmiotti, M. Smith, and J. Ragusa. Software design of sharp. *Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M and C + SNA 2007*, 2007. cited By 1.

Presents SHARP (Simulation-based High-efficiency Advanced Reactor Prototyping), a coupled multi-physics code framework for reactor simulation, from Argonne National Laboratory. It has a lightweight modular architecture, supporting non-restrictive data coupling via MOAB interface.

Frameworks in Chemistry and Chemical Engineering

[78] C.R. Jacob, S.M. Beyhan, R.E. Bulo, A.S.P. Gomes, A.W. Götz, K. Kiewisch, J. Sikkema, and L. Visscher. Pyadf - a scripting framework for multiscale quantum chemistry. *Journal of Computational Chemistry*, 32(10):2328–2338, 2011. cited By 15.

Presents PyADF, a scripting framework to automate workflows in multiscale quantum chemistry, from Karlsruhe Institute of Technology. It provides an extendable object-oriented hierarchy and allows writing Python scripts to integrate data and calculations.

[79] R. Morales-Rodriguez and R. Gani. Computer-aided multiscale modelling for chemical process engineering. *Computer Aided Chemical Engineering*, 24:207–212, 2007. cited By 10.

Presents a framework that integrates modeling and simulation for multiscale product-process simulation, from Technical University of Denmark. It facilitates a model-based approach with model analysis, generation and translation with CAPE-OPEN-compliant tools ModDev and MoT within ICAS. It further uses COM-based integration of the models with ICAS as simulator and EXCEL as orchestrator.

[80] R. Morales-Rodríguez, R. Gani, S. Déchelotte, A. Vacher, and O. Baudouin. Use of cape-open standards in the interoperability between modelling tools (mot) and process simulators (simulis®thermodynamics and prosimplus). *Chemical Engineering Research and Design*, 86(7):823–833, 2008. cited By 12.

Presents an integration approach for chemical process engineering using CAPE-OPEN, from CAPEC, Denmark. It demonstrates two case studies, where they integrate modeling and simulation tools first using a DLL interface for data flow, and secondly using an XML-based communication via COM.

[81] J. Morbach, A. Yang, and W. Marquardt. Ontocape-a large-scale ontology for chemical process engineering. *Engineering Applications of Artificial Intelligence*, 20(2):147–161, 2007. cited By 73.

Presents OntoCAPE, a large-scale ontology for chemical process engineering, from RWTH Aachen. It is used as a basis for semantic integration of multiscale chemical process simulations.

[82] G. Schopfer, A. Yang, L. von Wedel, and W. Marquardt. Cheops: A tool-integration platform for chemical process modelling and simulation. *International Journal on Software Tools for Technology Transfer*, 6(3):186–202, 2004. cited By 25.

Presents Component-Based Hierarchical Explorative Open Process Simulator (CHEOPS), an integration platform for chemical process modelling and simulation, from RWTH Aachen. It relies on a component-based architecture to wrap tools with common interfaces and integrate via CORBA. The wrappers are implemented in various languages including C++ and Python, compliant with CORBA.

[83] I. Thomas. A process unit modeling framework within a heterogeneous simulation environment. *Computer Aided Chemical Engineering*, 29:31–35, 2011. cited By 3.

Presents the Process Units Modeling Framework, a declarative environment for multiscale chemical process simulations, from Linde Engineering. It provides a declarative language for expressing the simulations in an implementation-independent manner, and maps this into a virtual machine, which binds (via COM, DLLs or static linking) the specifications with various tools. It provides an API for C/C++/Fortran.

[84] H.L. Woodcock, B.T. Miller, M. Hodoscek, A. Okur, J.D. Larkin, J.W. Ponder, and B.R. Brooks. Mscale: A general utility for multiscale modeling. *Journal of Chemical Theory and Computation*, 7(4):1208–1219, 2011. cited By 10.

Presents MSCALE, a multiscale coupling extension to the CHARMM macromolecular simulation platform. MSCALE organizes subsystems for each coupled simulation with implementing several hooks in the system (written in Fortran95) and uses MPI for communication.

[85] H. Xiong, Z. Huang, Z. Wu, and P.P. Conway. A generalized computational interface for combined thermodynamic and kinetic modeling. *Calphad: Computer Coupling of Phase Diagrams and Thermochemistry*, 35(3):391–395, 2011. cited By 7.

Presents an integrated framework for combined thermodynamic and kinetic modeling, from Sun Yat-sen University. It supports C, Fortran, Python, Java and Matlab programs, all through their DLL interfaces, while using thermodynamic databases. It further connects to Comsol through its Java interface.

[86] Y. Zhao, C. Jiang, and A. Yang. Towards computer-aided multiscale modelling: A generic supporting environment for model realization and execution. *Computers and Chemical Engineering*, 40:45–57, 2012. cited By 1.

Presents a generic supporting environment for multiscale model realization and execution, from University of Surrey. It proposes ontology-based (their own ontology integrated with ONTO-CAPE) conceptual modelling at the core of the framework, and use a Simulation Script Generator, to automatically generate executable code from the models. It further facilitates a component-based architecture for the generated codes, with CORBA/IDL standards.

[87] S.E. Zitney. Process/equipment co-simulation for design and analysis of advanced energy systems. *Computers and Chemical Engineering*, 34(9):1532–1542, 2010. cited By 19.

Presents Advanced Process Engineering Co-Simulator (APECS), a co-simulation framework for the design, analysis and optimization of energy plants, from National Energy Technology Laboratory, U.S. Department of Energy. It is built on CAPE-OPEN (CO) standard for plug-and-play interoperability - via COM/CORBA - with ANSYS Engineering Knowledge Manager for workflow/data management, and interfaces with many CO-compliant process simulation.

Frameworks in Material Science

[88] A. Bender, A. Poschlad, S. Bozic, and I. Kondov. A service-oriented framework for integration of domain-specific data models in scientific workflows. *Procedia Computer Science*, 18:1087–1096, 2013. cited By 2.

Presents a service-oriented language-independent platform for multiscale simulations for materials science on the grid and cloud, from Karlsruhe Institute of Technology. It follows a model-based approach: metamodeling of the simulation workflow with Eclipse, and an automatic transformation to generate the RESTful Web Services code to be deployed. It further proposes a REST/WSDL-based integration compatible with over 40 language and platform bindings.

[89] S. Bozic, I. Kondov, V. Meded, and W. Wenzel. Unicore based workflows for the simulation of organic light-emitting diodes. *UNICORE Summit 2012, Proceedings*, 15:15–25, 2012. cited By 4.

Presents a framework for the simulation of multiscale whole-device simulations, from Karlsruhe Institute of Technology. It facilitates the UNICORE grid framework/GridBeans technology as the communication and application wrapper infrastructure, while using data exchange based on Chemical Markup Language.

[90] M.J. Buehler, J. Dodson, A.C.T. Van Duin, P. Meulbroek, and W.A. Goddard III. The computational materials design facility (cmdf): A powerful framework for multi-paradigm multi-scale simulations. *Materials Research Society Symposium Proceedings*, 894:327–332, 2006.

Presents The Computational Materials Design Facility (CMDF), a framework for multiscale modeling and simulation of materials, from California Institute of Technology. It follows a data-driven approach, where a central Extended OpenBabel (XOB) data format, together with Python scripts, is used to glue simulation codes.

[91] M. Doi. Material modeling platform. *Journal of Computational and Applied Mathematics*, 149(1):13–25, 2002. cited By 16.

Presents a simulation platform, which provides a common interface to multiphysics models, from Nagoya University. It relies on a self-descriptive common data format (User Definable Format), a graphical handler (GOURMET) with customizable wrappers for different programming languages and simulation engines. The complete platform has later been renamed as Open Computational Tool for Advanced Material Technology [92].

[92] Masao Doi. Octa (open computational tool for advanced material technology). *Macromolecular Symposia*, 195(1):101–108, 2003.

[93] K. McManus, M. Cross, C. Walshaw, N. Croft, and A. Williams. Parallel performance in multi-physics simulation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2330 LNCS(PART 2):806–815, 2002.

Presents PHYSICA, a multiphysics computational framework for solidification/melting, from University of Greenwich. It is implemented in Fortran77 and relies on a modular architecture and MPI for integration.

[94] A.C.E. Reid, R.C. Lua, R.E. García, V.R. Coffman, and S.A. Langer. Modelling microstructures with oof2. *International Journal of Materials and Product Technology*, 35(3-4):361–373, 2009. cited By 35.

Presents OOF2, an object-oriented framework for finite element analysis of microstructures, from National Institute of Standards and Technology. It is written in C++/Python, and is extendable through a combination of C++/Python code and code wrappers for gluing.

[95] G.J. Schmitz and U. Prahl. Toward a virtual platform for materials processing. *JOM*, 61(5):19–23, 2009. cited By 18.

Presents AixVipMap®(the Aachen Virtual Platform for Materials Processing), a platform for integrated computational materials engineering, from RWTH Aachen. It follows a data-centric approach where many commercial black-box simulation software are integrated with parsers and converters for I/O compatibility with the standard VTK format.

[96] M.R. Tonks, D. Gaston, P.C. Millett, D. Andrs, and P. Talbot. An object-oriented finite element framework for multiphysics phase field simulations. *Computational Materials Science*, 51(1):20–29, 2012. cited By 27.

Frameworks in Environmental Science, Earth and Planetary Sciences

[97] D.P. Bacon, N.N. Ahmad, Z. Boybeyi, T.J. Dunn, M.S. Hall, P.C.S. Lee, R.A. Sarma, M.D. Turner, K.T. Waight III, S.H. Young, and J.W. Zack. A dynamically adapting weather and dispersion model: The operational multiscale environment model with grid adaptivity (omega). *Monthly Weather Review*, 128(7 I):2044–2076, 2000. cited By 101.

Presents The Operational Multiscale Environment Model with Grid Adaptivity (OMEGA), a framework for multiscale atmospheric simulation and weather prediction, developed by Science Applications International Corporation. It consists of modules of Fortran code.

[98] V Balaji. The flexible modeling system. In *Earth System Modelling-Volume 3*, pages 33–41. Springer, 2012.

Presents the Flexible Modeling System (FMS), a framework for atmospheric, oceanic, and climate system modeling, from Princeton University. Written in Fortran, it has a modular set of components with common interfaces and parallel I/O.

[99] M. T. Bettencourt. Distributed model coupling framework. In *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on*, pages 284–290. IEEE, 2002.

Presents a distributed model coupling framework for geophysical models, from Stennis Space Center. Written in C++, it couples models in C++ and Fortran through a CORBA infrastructure.

[100] A. Bodas-Salcedo, M.J. Webb, S. Bony, H. Chepfer, J.-L. Dufresne, S.A. Klein, Y. Zhang, R. Marchand, J.M. Haynes, R. Pincus, and V.O. John. Cosp: Satellite simulation software for model assessment. *Bulletin of the American Meteorological Society*, 92(8):1023–1043, 2011. cited By 84.

Presents the Cloud Feedback Model Intercomparison Project (CFMIP) Observation Simulator Package (COSP), an integrated satellite simulator, developed by Met Office Hadley Centre and other parties. It consists of Fortran90 interfaces to various cloud products.

[101] F. Branger, I. Braud, S. Debionne, P. Viallet, J. Dehotin, H. Henine, Y. Nedelec, and S. Anquetin. Towards multi-scale integrated hydrological models using the liquid®framework. overview of the concepts and first application examples. *Environmental Modelling and Software*, 25(12):1672–1681, 2010. cited By 19.

Presents LIQUID, a proprietary framework for integrated hydrological modeling from Hydrowide. It consists of C++ modules and wrappers for other programming languages; while providing templates for the development of new modules.

[102] S. Buis, A. Piacentini, and D. Déclat. Palm: A computational framework for assembling high-performance computing applications. *Concurrency Computation Practice and Experience*, 18(2):231–245, 2006. cited By 33.

Presents PALM (later open-source as OpenPALM), a framework for assembling high-performance computing applications in oceanography, from CERFACS. Written in C, it supports the instrumentation of code in C/C++/Fortran, organized in modular components dynamically coupled via MPI.

[103] D. Byun and K.L. Schere. Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (cmaq) modeling system. *Applied Mechanics Reviews*, 59(1-6):51–76, 2006. cited By 690.

Presents Community Multiscale Air Quality (CMAQ) modeling system, an open-source modular system linking meteorology, emissions, air quality and health effects. It uses a data-centric approach with netCDF and its Fortran/C API for data interoperability among various models.

[104] D. Byun and K. L. Schere. Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (cmaq) modeling system. *Applied Mechanics Reviews*, 59(2):51–77, 2006.

Presents the Community Multiscale Air Quality (CMAQ) modeling system which is an open source framework for conducting air quality model simulations, part of the Community Modeling and Analysis System developed by the U.S. Environment Protection Agency, Atmospheric Science Modeling Division. Written in Fortran90, it consists of scientific libraries and modules.

[105] D. Coumou, S. Matthäi, S. Geiger, and T. Driesner. A parallel fe-fv scheme to solve fluid flow in complex geologic media. *Computers and Geosciences*, 34(12):1697–1707, 2008. cited By 19.

Presents Complex System Modelling Platform (CSMP++), a freely available (binary-only) object-oriented framework for the simulation of earth systems, from ETH Zurich. Written in C++, it uses design-patterns, templates and polymorphism for extension. It supports MPI.

[106] A.P. Craig, R. Jacob, B. Kauffman, T. Bettge, J. Larson, E. Ong, C. Ding, and Y. He. Cpl6: The new extensible, high performance parallel coupler for the community climate system model. *International Journal of High Performance Computing Applications*, 19(3):309–328, 2005. cited By 43.

Presents cpl6, a coupler based which integrates several independent models into a climate system (Community Climate System Model). It is written in Fortran90 and based on Model Coupling Toolkit, extending it with coupling functions and interfaces.

[107] L. A. Drummond, J. Demmel, C. R. Mechoso, H. Robinson, K. Sklower, and J. A. Spahr. A data broker for distributed computing environments. In *Computational Science ICCS 2001*, pages 31–40. Springer, 2001.

Presents the Distributed Data Broker (DDB), a toolkit for the coupling of distributed multi-application computing systems evolved from a coupled atmosphere-ocean modeling system. It implements a plug-in communication infrastructure with the consumer-producer model, where applications register themselves and connect via the central broker.

[108] B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Müthing, P. Nuske, A. Tatomir, M. Wolff, and R. Helmig. Dumux: Dune for multi-phase,component,scale,physics,... flow and transport in porous media. *Advances in Water Resources*, 34(9):1102–1112, 2011. cited By 33.

Presents DuMux, an open-source simulator for flow and transport processes in porous media, based on the Distributed and Unified Numerics Environment (DUNE), developed mainly by University of Stuttgart. It is written in C++ and extends DUNE's idea of using generic programming techniques and slim interfaces to integrate legacy code or new applications.

[109] R.W. Ford, G.D. Riley, M.K. Bane, C.W. Armstrong, and T.L. Freeman. Gcf: A general coupling framework. *Concurrency Computation Practice and Experience*, 18(2):163–181, 2006. cited By 13.

> Presents the General Coupling Framework and its prototype implementation Bespoke Framework Generator, a framework for coupled earth system simulations, from University of Manchester. Written in Fortran, it provides a common interface, separate wrapping of Fortran simulations, XML-based metadata and MPI communication for interoperability.

[110] C.C. Goodrich, A.L. Sussman, J.G. Lyon, M.A. Shay, and P.A. Cassak. The cism code coupling strategy. *Journal of Atmospheric and Solar-Terrestrial Physics*, 66(15-16 SPEC. ISS.):1469–1479, 2004. cited By 9.

> Presents a code coupling framework for solar-terrestrial physics, from Center for Integrated Space Weather Modeling (CISM). It proposes InterComm as the infrastructure for parallel data communication, and Overture for scientific operations (interpolation, etc) for integration.

[111] J.B. Gregersen, P.J.A. Gijsbers, and S.J.P. Westen. Openmi: Open modelling interface. *Journal of Hydroinformatics*, 9(3):175–191, 2007. cited By 151.

> Presents Open Modelling Interface (OpenMI), a standard for integrating environmental models. It serves as common platform for structure and data exchange interfaces of models in C# or Java as source code or DLLs, achieving interoperability through compliance.

[112] O. Kolditz, S. Bauer, L. Bilke, N. Böttcher, J.O. Delfs, T. Fischer, U.J. Görke, T. Kalbacher, G. Kosakowski, C.I. McDermott, C.H. Park, F. Radu, K. Rink, H. Shao, H.B. Shao, F. Sun, Y.Y. Sun, A.K. Singh, J. Taron, M. Walther, W. Wang, N. Watanabe, Y. Wu, M. Xie, W. Xu, and B. Zehner. Opengeosys: An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (thm/c) processes in porous media. *Environmental Earth Sciences*, 67(2):589–599, 2012. cited By 72.

> Presents OpenGeoSys (OGS), an open-source framework for numerical simulation of thermo-hydro-mechanical-chemical processes in porous media, from Helmholtz Centre for Environmental Research. Evolved through Fortran, C and C++ implementations, currently it uses an object-oriented architecture for flexibility and extension.

[113] S. Kumar, C. Peters-Lidard, Y. Tian, R. Reichle, J. Geiger, C. Alonge, J. Eylander, and P. Houser. An integrated hydrologic modeling and data assimilation framework. *Computer*, 41(12):52–59, 2008. cited By 13.

Presents The Land Information System (LIS), a multiscale hydrologic modeling and data assimilation framework, from NASA Goddard Space Flight Center. Together with a toolchain including The Land Data Toolkit and Land Verification Toolkit, it allows the integration of observational data from many sources with modeling, simulation, prediction and optimization components. It relies on user-defined functions and polymorphism for extension and integration.

[114] D. Lam, L. Leon, S. Hamilton, N. Crookshank, D. Bonin, and D. Swayne. Multi-model integration in a decision support system: A technical user interface approach for watershed and lake management scenarios. *Environmental Modelling and Software*, 19(3):317–324, 2004. cited By 25.

Presents RAISON (Regional Analysis by Intelligent Systems ON microcomputers), a multimodel simulation framework for watershed management, from Environment Canada. Written in C/C++/Visual Basic, it follows a data-centric approach, integrating GIS and various models in databases in a decision support context.

[115] J. Larson, R. Jacob, and E. Ong. The model coupling toolkit: A new fortran90 toolkit for building multiphysics parallel coupled models. *International Journal of High Performance Computing Applications*, 19(3):277–292, 2005. cited By 94.

Presents the Model Coupling Toolkit (MCT), a framework that simplifies the construction of parallel coupled multiphysics models for climate, developed by Argonne National Laboratory. Written in Fortran90, it facilitates object-orientation and maintains a hierarchy of classes.

[116] G.H. Leavesley, S.L. Markstrom, P.J. Restrepo, and R.J. Viger. A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modelling. *Hydrological Processes*, 16(2):173–187, 2002. cited By 78.

Presents modular modelling system (MMS), an open-source modular modelling framework for multidisciplinary distributed hydrological models, from US Geological Survey. It promotes highly modular development of models and follows a database-centric integration.

[117] K.-A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H.M. Nilsen, and B. Skaflestad. Open-source matlab implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322, 2012. cited By 57.

Presents Matlab Reservoir Simulation Toolbox (MRST), an open source framework for rapid prototyping and evaluation of reservoir simulations, from SINTEF Applied Mathematics. It maintains a modular Matlab framework that simplifies the development of new applications.

[118] X.-J. Luo, T. Stylianopoulos, V.H. Barocas, and M.S. Shephard. Multiscale computation for bioartificial soft tissues with complex geometries. *Engineering with Computers*, 25(1):87–95, 2009. cited By 7.

Presents a multiscale component toolkit (MCTK), a multiscale extension of the Trellis framework for soft tissue simullation, from Rensselaer Polytechnic Institute. Written in C++, it relies on object-oriented mechanisms and MPI-based communication for integration and extension.

[119] J. Michalakes, J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang. The weather research and forecast model: Software architecture and performance. *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing in Meteorology*, pages 156–168, 2005. cited By 75.

Presents the Weather Research and Forecast (WRF), a multiscale modeling system for atmospheric research, from U.S. National Center for Atmospheric Research. It has a layered architecture, using MPI-/OpenMP-based communication and data conversions among models. It facilitates a registry database holding data structures and metadata, from which code is automatically generated to couple models.

[120] P. Montarnal, A. Dimier, E. Deville, E. Adam, J. Gaombalet, A. Bengaouer, L. Loth, and C. Chavant. Coupling methodology within the software platform alliances. *CoRR*, abs/cs/0611127, 2006.

Presents Alliances, a software platform for the simulation of nuclear waste storage and disposal repository, from CEA/ANDRA/EDF France. It is written in Python and relies on wrappers for integration of legacy code in other languages such as C and Fortran. It is integrated into the Salome platform.

[121] M. Peszynska, Q. Lu, and M.F. Wheeler. Multiphysics coupling of codes. *Computational methods in water resources - Volume 1 - Computational methods for subsurface flow and transport*, pages 175–182, 2000. cited By 14.

Presents The Implicit Parallel Accurate Reservoir Simulator (IPARS), a code coupling framework for the simulation of multi-component multiphase flow problems, from The University of Texas at Austin. It is written mainly in Fortran77 and partly in C, and relies on user defined functions and a driver to couple codes.

[122] S. Popinet. Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comp. Phys*, 190:572–600, 2003.

Presents Gerris, an open-source framework for geophysical fluid dynamics from National Institute of Water and Atmospheric Research New

Zealand. It is written in C and relies on an emulated class/inheritance mechanism for modularity and extension.

[123] R. Quilbé and A.N. Rousseau. Gibsi: An integrated modelling system for watershed management - sample applications and current developments. *Hydrology and Earth System Sciences*, 11(6):1785–1795, 2007. cited By 16.

Presents GIBSI, an integrated multiscale modelling system for watershed management, from INRS-ETE. It follows a data-centric approach, integrating GIS and various models in databases in a decision support context.

[124] R. Redler, S. Valcke, and H. Ritzdorf. Oasis4 - a coupling software for next generation earth system modelling. *Geoscientific Model Development*, 3(1):87–104, 2010. cited By 28.

Presents OASIS4 (Ocean Atmosphere Sea Ice Soil) coupling software for multiscale earth system modelling, from CERFACS. Written in Fortran90 and C, it has a central driver unit, communicating with other components through a common API via MPI, and using XML-based description/configuration files for coupling. It is compatible with Common Information Model (CIM) metadata for coupling configuration.

[125] E. Tan, E. Choi, P. Thoutireddy, M. Gurnis, and M. Aivazis. Geoframework: Coupling multiple models of mantle convection within a computational framework. *Geochemistry, Geophysics, Geosystems*, 7(6), 2006. cited By 66.

Presents GeoFramework, a coupling framework for geophysical solvers, from California Institute of Technology. Written in Python, it extends the Pyre framework; it provides Python bindings for solver code in C/C++/Fortran and coupling via Python.

[126] G. Tóth, B. van der Holst, I.V. Sokolov, D.L. De Zeeuw, T.I. Gombosi, F. Fang, W.B. Manchester, X. Meng, D. Najib, K.G. Powell, Q.F. Stout, A. Glocer, Y.-J. Ma, and M. Opher. Adaptive numerical algorithms in space weather modeling. *Journal of Computational Physics*, 231(3):870–903, 2012. cited By 87.

Presents the Space Weather Modeling Framework (SWMF), a open source common framework for various modeling components for space weather, developed by University of Michigan. It consists of core Fortran90 modules, and Perl scripts, integrated via MPI.

[127] F. Villa. Integrating modelling architecture: A declarative framework for multi-paradigm, multi-scale ecological modelling. *Ecological Modelling*, 137(1):23–42, 2001. cited By 32.

Presents Integrated Modelling Architecture(IMA), a declarative framework and an open-source software toolkit (Integrated Modelling Toolkit) to allow integrated meta-modelling for ecology, from University of Maryland. It relies on a generic XML specification of model components and integration of existing models through this common language. The toolkit is a discontinued alpha implementation in C++.

[128] M. Xue, K.K. Droegemeier, and V. Wong. The advanced regional prediction system (arps) - a multi-scale nonhydrostatic atmospheric simulation and prediction model. part i: Model dynamics and verification. *Meteorology and Atmospheric Physics*, 75(3-4):161–193, 2000. cited By 501.

Presents the Advanced Regional Prediction System (ARPS), an open source multi-scale atmospheric simulation/prediction system developed at the University of Oklahoma. The system is written in Fortran-77/90 and consists of self-contained and extendible modules.

Frameworks in Life and Health Sciences

[129] A. Arbona, S. Benkner, G. Engelbrecht, J. Fingberg, M. Hofmann, K. Kumpf, G. Lonsdale, and A. Woehrer. A service-oriented grid infrastructure for biomedical data and compute services. *IEEE Transactions on Nanobioscience*, 6(2):136–141, 2007. cited By 17.

Presents the EU project @neurIST, which aims to develop a service-oriented infrastructure for multiscale analysis and integration of biomedical data and services. It supports semantic interoperability through a service-oriented architecture, in terms of web service standards and several ontologies.

[130] C. Bajaj, A. Dicarlo, and A. Paoluzzi. Proto-plasm: Parallel language for adaptive and scalable modelling of biosystems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1878):3045–3065, 2008. cited By 3.

Presents Proto-Plasm, a framework for multiscale geometric modeling and multiphysics simulation of biosystems. It provides an integrated development environment to model the geometry using the PLASM symbolic language (an extension of FL, embedded in Erlang), integrating with CellML and SBML.

[131] D.A. Beard, M.L. Neal, N. Tabesh-Saleki, C.T. Thompson, J.B. Bassingtwaighte, M. Shimoyama, and B.E. Carlson. Multiscale modeling and data integration in the virtual physiological rat project. *Annals of Biomedical Engineering*, 40(11):2365–2378, 2012. cited By 12.

Presents The Virtual Physiological Rat (VPR) Project from National Institute of General Medical Sciences (NIGMS). VPS involves a multi-scale model integration platform, which uses SemGen for the semantic integration of various models (such as CellML, SBML and MML) using ontologies (such as Gene Ontology and BPO) and can automatically generate interoperable SemSim models.

[132] C. Bradley, A. Bowery, R. Britten, V. Budelmann, O. Camara, R. Christie, A. Cookson, A.F. Frangi, T.B. Gamage, T. Heidlauf, S. Krittian, D. Ladd, C. Little, K. Mithraratne, M. Nash, D. Nickerson, P. Nielsen, T. Nordbø, S. Omholt, A. Pashaei, D. Paterson, V. Rajagopal, A. Reeve, O. Röhrle, S. Safaei, R. Sebastián, M. Steghöfer, T. Wu, T. Yu, H. Zhang, and P. Hunter. Opencmiss: A multi-physics & multi-scale computational infrastructure for the vph/physiome project. *Progress in Biophysics and Molecular Biology*, 107(1):32–47, 2011. cited By 29.

Presents OpenCMISS, an open-source flexible framework for the simulation of coupled multiphysics problems in the VPH/Physiome Project. Written in Fortran95/2003, it has an object-based architecture with support for Fortran/C bindings and SWIG bindings to C++/Python. Tt relies on data structures compatible with DSLs such as CellML and FieldML for interoperability.

[133] S.U. Chaudhary, S.-Y. Shin, D. Lee, J.-H. Song, and K.-H. Cho. Elecans - an integrated model development environment for multiscale cancer systems biology. *Bioinformatics*, 29(7):957–959, 2013. cited By 1.

Presents ELECANS (electronic cancer system), an integrated modeling environment for multiscale cancer systems biology, from Korea Advanced Institute of Science and Technology. Written in C#, it has a layered architecture for integrating third-party agent-based simulations through DLL bindings to the environment.

[134] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, and D.G. Thelen. Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 54(11):1940–1950, 2007. cited By 480.

Presents OpenSIM, an open-source platform for multiscale neuromuscoloskeletal simulations, from Stanford University. Written in C++/Java, it has a plug-in architecture with object-oriented extensions and DLLs for integration of low-level computational components.

[135] B. Drawert, S. Engblom, and A. Hellander. Urdme: A modular framework for stochastic simulation of reaction-transport processes in complex geometries. *BMC Systems Biology*, 6, 2012. cited By 14.

Presents URDME, a flexible software framework for general stochastic reaction-transport modeling and simulation, from University of California and other institutions. It has its core functionality implemented in C; and interfaces with COMSOL Multiphysics,and Matlab for integration and extra features such as postprocessing. It also supports SBML.

[136] E.Z. Erson and M.C. Cavusoglu. A software framework for multiscale and multilevel physiological model integration and simulation. *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology"*, pages 5449–5453, 2008. cited By 3.

Presents a software framework for the integration of multiscale physiological processes, from Case Western Reserve University. It has a layered architecture separates conceptual levels, and uses modeling languages (e.g. MML) and ontologies (e.g. Foundational Model of Anatomy) to provide semantic mappings among components via the link layer.

[137] M. Feig, J. Karanicolas, and C.L. Brooks III. Mmtsb tool set: Enhanced sampling and multiscale modeling methods for applications in structural biology. *Journal of Molecular Graphics and Modelling*, 22(5):377–395, 2004. cited By 381.

Presents the Multiscale Modeling Tools for Structural Biology (MMTSB) Tool Set, a set of utilities and libraries for multiscale modeling and simulation of proteins and nucleic acids, developed by Scripps Research Institute. The system is written as Perl modules interfacing with CHARMM and Amber and MONSSTER packages.

[138] A. Garny, D.P. Nickerson, J. Cooper, R.W.D. Santos, A.K. Miller, S. McKeever, P.M.F. Nielsen, and P.J. Hunter. Cellml and associated tools and techniques. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1878):3017–3043, 2008. cited By 55.

Presents CellML, developed by University of Auckland and other institutions, an open standard XML-based markup language to describe multiphysics and multiscale cell model and also discusses various tools including editor, validator, code generator and validators for CellML.

[139] J.H. Gennari, M.L. Neal, B.E. Carlson, and D.L. Cook. Integration of multi-scale biosimulation models via light-weight semantics, 2008. cited By 17.

Presents an ontological framework for the integration of biosimulation models, from University of Washington. It aims to develop an Application Model Ontology (AMO), in order to annotate legacy code/models with semantics to identify mappings, merge and eventually generate code.

[140] D. Han, Q. Liu, and Q. Luo. China physiome project: A comprehensive framework for anatomical and physiological databases from the china digital human and the visible rat. *Proceedings of the IEEE*, 97(12):1969–1976, 2009. cited By 6.

Presents the Chinese Physiome Project, which aims to develop a multiscale modeling framework for computational physiology. It follows a model-based approach:,using an ontology based on the standard controlled vocabulary MeSH, and a multiscale modeling markup language (M3L) capable of expressing model, data and web service deployment information.

[141] J. Hetherington, I.D.L. Bogle, P. Saffrey, O. Margoninski, L. Li, M.V. Rey, S. Yamaji, S. Baigent, J. Ashmore, K. Page, R.M. Seymour, A. Finkelstein, and A. Warner. Addressing the challenges of multiscale model management in systems biology. *Computers and Chemical Engineering*, 31(8):962–979, 2007. cited By 14.

Presents a framework for integrating models and tools for multiscale systems biology within The UCL (University College London) Beacon Project. Written in C++, it has a component-based architecture supporting a variety of XML-based specification languages (e.g. Composite Model Description Language), native XML databases, and wrappers for simulation tools such as Mathematica.

[142] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J. Doyle, and H. Kitano. The erato systems biology workbench: enabling interaction and exchange between software tools for computational biology. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 450–461, 2002. cited By 46.

Presents the ERATO Systems Biology Workbench (SBW), an open-source software framework for multiscale and multitheoretic simulations in systems biology, from California Institute of Technology. Written in Java, it provides a broker-based communication infrastructure for interoperability through C, C++, Java, Delphi and Python bindings and a common API.

[143] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J.C. Doyle, H. Kitano, A.P. Arkin, B.J. Bornstein, D. Bray, A. Cornish-Bowden, A.A. Cuellar, S. Dronov, E.D. Gilles, M. Ginkel, V. Gor, I.I. Goryanin, W.J. Hedley, T.C. Hodgman, J.-H. Hofmeyr, P.J. Hunter, N.S. Juty, J.L. Kasberger, A. Kremling, U. Kummer, N. Le Novére, L.M. Loew, D. Lucio, P. Mendes, E. Minch, E.D. Mjolsness, Y. Nakayama, M.R. Nelson, P.F. Nielsen, T. Sakurada, J.C. Schaff, B.E. Shapiro, T.S. Shimizu, H.D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (sbml): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003. cited By 1269.

Presents the systems biology markup language (SBML), a free and open interchange format for computer models of biological processes, developed by California Institute of Technology and other institutions. It provides a common language to be used by numerous tools for analysis, utility, etc.

[144] P. Hunter, P.V. Coveney, B. De Bono, V. Diaz, J. Fenner, A.F. Frangi, P. Harris, R. Hose, P. Kohl, P. Lawford, K. McCormack, M. Mendes, S. Omholt, A. Quarteroni, J. Skår, J. Tegner, S.R. Thomas, I. Tollis, I. Tsamardinos, J.H.G.M. Van Beek, and M. Viceconti. A vision and strategy for the virtual physiological human in 2010 and beyond. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1920):2595–2614, 2010. cited By 66.

Presents the Virtual Physiological Human (VPH) project and toolkit for the complete multiscale computational modeling of the human body, supported by EU projects and initiatives. It incorporates many standards, ontologies, markup and modeling languages, data conversion utilities and many other tools in an integrated framework for a high level of interoperability. It is a joint effort with the Physiome Initiative.

[145] J.A. Izaguirre, R. Chaturvedi, C. Huang, T. Cickovski, J. Coffland, G. Thomas, G. Forgacs, M. Alber, G. Hentschel, S.A. Newman, and J.A. Glazier. Compucell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, 20(7):1129–1137, 2004. cited By 96.

Presents Compu-Cell (later renamed as CompuCell3D), an open-source multi-model software framework for the morphogenesis simulation. Written in C++, it relies on its object-oriented architecture, design patterns and a XML-based domain-specific language (Biologo, CompuCell3D Markup Language) for extensibility and coupling.

[146] S.P. Kumar and J.C. Feidler. Biospice: A computational infrastructure for integrative biology. *OMICS A Journal of Integrative Biology*, 7(3):225, 2003. cited By 13.

Presents BioSPICE (Biological Simulation Program for Intra-Cellular Evaluation), an open source framework and software toolset for Systems Biology, from The University of Texas-Houston Medical School. Written in Java, it maintains a DashBoard, based on Open Agent Architecture and NetBeans software technologies, to allow data integration of models/simulations. It also supports various technologies such as SBML.

[147] F. Mitha, T.A. Lucas, F. Feng, T.B. Kepler, and C. Chan. The multiscale systems immunology project: Software for cell-based immunological simulation. *Source Code for Biology and Medicine*, 3, 2008. cited By 12.

Presents The Multiscale Systems Immunology (MSI), an object-oriented. modular simulation framework for immune system simulations, from Duke University. It has two versions, one written in Python for flexibility and the other in C++ for performance; using wrappers to bridge two. It relies on parametrization, factory design pattern and inheritance to extend the framework.

[148] M.L. Neal, J.H. Gennari, T. Arts, and D.L. Cook. Advances in semantic representation for multiscale biosimulation: A case study in merging models. *Pacific Symposium on Biocomputing 2009, PSB 2009*, pages 304–315, 2009. cited By 14.

Presents a framework for semantic integration of multiscale biosimulation models, from University of Washington. It uses a semantic framework (SemSim) and ontologies (e.g. Ontology of Physics for Biology and Foundational Model of Anatomy), to integrate models in various languages such as SBML, CellML and MML. The integration of Matlab and JSim is achieved through automatic code generation from the models; while the reverse direction is also done to lift the code representation to the declarative model as well.

[149] E. Neufeld, D. Szczerba, N. Chavannes, and N. Kuster. A novel medical image data-based multi-physics simulation platform for computational life sciences. *Interface Focus*, 3(2), 2013. cited By 1.

Presents a medical data-based coupling framework for multiscale simulation of biological systems, from Foundation for Research on Information Technologies in Society. It relies on the common data formats such as VTK and XDMF, and Python scripting to orchestrate the integration.

[150] J. Pitt-Francis, P. Pathmanathan, M.O. Bernabeu, R. Bordas, J. Cooper, A.G. Fletcher, G.R. Mirams, P. Murray, J.M. Osborne, A. Walter, S.J. Chapman, A. Garny, I.M.M. van Leeuwen, P.K. Maini, B. Rodríguez, S.L. Waters, J.P. Whiteley, H.M. Byrne, and D.J. Gavaghan. Chaste: A test-driven approach to software development for biological modelling. *Computer Physics Communications*, 180(12):2452–2471, 2009. cited By 86.

Presents Chaste (Cancer, heart and soft-tissue environment), a open source generic library for computational simulations in biology, developed by University of Oxford. Written in C++, it uses object-oriented inheritance for extension. Furthermore, it has facilities for code generation from CellML models.

[151] J. Rasinmäki, A. Mäkinen, and J. Kalliovirta. Simo: An adaptable simulation framework for multiscale forest resource data. *Computers and Electronics in Agriculture*, 66(1):76–84, 2009. cited By 21.

Presents SIMO, an open-source framework for forest management planning, from University of Helsinki. Written mainly in Python, it heavily uses XML-based descriptions for internal data structure, data mappings, model, simulation parameters, etc. to aid interoperability.

[152] J. Starruß, W. De Back, L. Brusch, and A. Deutsch. Morpheus: A user-friendly modeling environment for multiscale and multicellular systems biology. *Bioinformatics*, 30(9):1331–1332, 2014. cited By 3.

Presents Morpheus, an environment for the multiscale simulation and integration of cell-based models, from Technische Universität Dresden. It uses domain-specific languages (e.g. SBML) to abstract biological and mathematical models; and allows model-based integration, simulation and visualization.

[153] T. Sütterlin, C. Kolb, H. Dickhaus, D. Jäger, and N. Grabe. Bridging the scales: Semantic integration of quantitative sbml in graphical multi-cellular models and simulations with episim and copasi. *Bioinformatics*, 29(2):223–229, 2013. cited By 6.

Presents EPISIM, a platform for semantic integration of SBML and cell behavioural models, from Heidelberg University. It can semantically link SBML-based models with graphical cell process models and generate execution code. For this, it uses techniques from Model-Driven Engineering such as model transformation.

[154] Y. Suzuki, Y. Asai, H. Oka, E. Heien, T. Urai, T. Okamoto, Y. Yumikura, K. Tominaga, Y. Kido, M. Nakanishi, K. Hagihara, Y. Kurachi, and T. Nomura. A platform for in silico modeling of physiological systems iii. *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pages 2803–2806, 2009. cited By 3.

Presents an open platform for multiscale physiological modeling, including insilicoML(ISML) and insilicoIDE(ISIDE), from Osaka University. ISML acts as a common data model and are translated in ISIDE into C++ code to be simulated. The platform further complies with CellML and MathML.

[155] S.R. Thomas, P. Baconnier, J. Fontecave, J.-P. Françoise, F. Guillaud, P. Hannaert, A. Hernández, V. Le Rolle, P. Maziére, F. Tahi, and R.J. White. Saphir: A physiome core model of body fluid homeostasis and blood pressure regulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1878):3175–3197, 2008. cited By 26.

Presents a Systems Approach for PHysiological Integration of Renal, cardiac and respiratory function (SAPHIR) as a multi-scale modelling environment within the Physiome [XX] project. It merits the same methodology of the project for interoperability.

Science Reports **Department of Mathematics and Computer Science**
Technische Universiteit Eindhoven

If you want to receive reports, send an email to: wsinsan@tue.nl (we cannot guarantee the availability of the requested reports).

In this series appeared (from 2012):

12/01	S. Cranen	Model checking the FlexRay startup phase
12/02	U. Khadim and P.J.L. Cuijpers	Appendix C / G of the paper: Repairing Time-Determinism in the Process Algebra for Hybrid Systems ACP
12/03	M.M.H.P. van den Heuvel, P.J.L. Cuijpers, J.J. Lukkien and N.W. Fisher	Revised budget allocations for fixed-priority-scheduled periodic resources
12/04	Ammar Osaiweran, Tom Fransen, Jan Friso Groote and Bart van Rijnsoever	Experience Report on Designing and Developing Control Components using Formal Methods
12/05	Sjoerd Cranen, Jeroen J.A. Keiren and Tim A.C. Willemse	A cure for stuttering parity games
12/06	A.P. van der Meer	CIF MSOS type system
12/07	Dirk Fahland and Robert Prüfer	Data and Abstraction for Scenario-Based Modeling with Petri Nets
12/08	Luc Engelen and Anton Wijs	Checking Property Preservation of Refining Transformations for Model-Driven Development
12/09	M.M.H.P. van den Heuvel, M. Behnam, R.J. Bril, J.J. Lukkien and T. Nolte	Opaque analysis for resource-sharing components in hierarchical real-time systems - extended version –
12/10	Milosh Stolikj, Pieter J. L. Cuijpers and Johan J. Lukkien	Efficient reprogramming of sensor networks using incremental updates and data compression
12/11	John Businge, Alexander Serebrenik and Mark van den Brand	Survival of Eclipse Third-party Plug-ins
12/12	Jeroen J.A. Keiren and Martijn D. Klabbers	Modelling and verifying IEEE Std 11073-20601 session setup using mCRL2
12/13	Ammar Osaiweran, Jan Friso Groote, Mathijs Schuts, Jozef Hooman and Bart van Rijnsoever	Evaluating the Effect of Formal Techniques in Industry
12/14	Ammar Osaiweran, Mathijs Schuts, and Jozef Hooman	Incorporating Formal Techniques into Industrial Practice
13/01	S. Cranen, M.W. Gazda, J.W. Wesselink and T.A.C. Willemse	Abstraction in Parameterised Boolean Equation Systems
13/02	Neda Noroozi, Mohammad Reza Mousavi and Tim A.C. Willemse	Decomposability in Formal Conformance Testing
13/03	D. Bera, K.M. van Hee and N. Sidorova	Discrete Timed Petri nets
13/04	A. Kota Gopalakrishna, T. Ozcelebi, A. Liotta and J.J. Lukkien	Relevance as a Metric for Evaluating Machine Learning Algorithms
13/05	T. Ozcelebi, A. Weffers-Albu and J.J. Lukkien	Proceedings of the 2012 Workshop on Ambient Intelligence Infrastructures (WAmIi)
13/06	Lotfi ben Othmane, Pelin Angin, Harold Weffers and Bharat Bhargava	Extending the Agile Development Process to Develop Acceptably Secure Software
13/07	R.H. Mak	Resource-aware Life Cycle Models for Service-oriented Applications managed by a Component Framework
13/08	Mark van den Brand and Jan Friso Groote	Software Engineering: Redundancy is Key
13/09	P.J.L. Cuijpers	Prefix Orders as a General Model of Dynamics

14/01	Jan Friso Groote, Remco van der Hofstad and Matthias Raffelsieper	On the Random Structure of Behavioural Transition Systems
14/02	Maurice H. ter Beek and Erik P. de Vink	Using mCRL2 for the analysis of software product lines
14/03	Frank Peeters, Ion Barosan, Tao Yue and Alexander Serebrenik	A Modeling Environment Supporting the Co-evolution of User Requirements and Design
14/04	Jan Friso Groote and Hans Zantema	A probabilistic analysis of the Game of the Goose
14/05	Hrishikesh Salunkhe, Orlando Moreira and Kees van Berkel	Buffer Allocation for Real-Time Streaming on a Multi-Processor without Back-Pressure
14/06	D. Bera, K.M. van Hee and H. Nijmeijer	Relationship between Simulink and Petri nets
14/07	Reinder J. Bril and Jinkyu Lee	CRTS 2014 - Proceedings of the 7th International Workshop on Compositional Theory and Technology for Real-Time Embedded Systems
14/08	Fatih Turkmen, Jerry den Hartog, Silvio Ranise and Nicola Zannone	Analysis of XACML Policies with SMT
14/09	Ana-Maria Şutîi, Tom Verhoeff and M.G.J. van den Brand	Ontologies in domain specific languages – A systematic literature review
14/10	M. Stolikj, T.M.M. Meyfroyt, P.J.L. Cuijpers and J.J. Lukkien	Improving the Performance of Trickle-Based Data Dissemination in Low-Power Networks
15/01	Önder Babur, Tom Verhoeff and Mark van den Brand	Multiphysics and Multiscale Software Frameworks: An Annotated Bibliography
15/02	Various	Proceedings of the First International Workshop on Investigating Dataflow In Embedded computing Architectures (IDEA 2015)
15/03	Hrishikesh Salunkhe, Alok Lele, Orlando Moreira and Kees van Berkel	Buffer Allocation for Realtime Streaming Applications Running on a Multi-processor without Back-pressure
15/04	J.G.M. Mengerink, R.R.H. Schiffelers, A. Serebrenik, M.G.J. van den Brand	Evolution Specification Evaluation in Industrial MDSE Ecosystems
15/05	Sarmen Keshishzadeh and Jan Friso Groote	Exact Real Arithmetic with Pertubation Analysis and Proof of Correctness
15/06	Jan Friso Groote and Anton Wijs	An $O(m \log n)$ Algorithm for Stuttering Equivalence and Branching Bisimulation