Controlling the Design Dynamics of Oil and Gas Pipelines for Effective Fluid Flow Using Domain Specific Modelling

*[1]Japheth, B. R., [2]Osaisai, E. F. and [2]Orukari, M.
[1]Department of Computer Science, Niger Delta University Wilberforce Island Bayelsa State, Nigeria.
[2]Department of Mathematics, Niger Delta University Wilberforce Island Bayelsa State, Nigeria.

Article Information

Article # 02021
Received: 15th June 2021
Revision: 13th Nov. 2021
Acceptance: 24th Nov. 2021
Published:25thNov. 2021

Key Words
Potential design, salient Characteristics, pipeline Systems, conceptual linkages, Pipe and fluid flow

Abstract

In the oil and gas industry in Nigeria, there have been constant demands for more cost-effective and efficient tools to productive pipeline design. After a careful examination, it became evident, that, stakeholders in the oil and gas pipeline industry are faced with the problems of model selection, cost effectiveness in routing designs, and project quality controls. These factors so far have hindered economic recovery in the industry. This paper, therefore, is to address these demands by coming up with a software solution with type-systems and semantics under domain specific modelling, that can simplify design and implementation of oil and gas transmission pipeline structures. Addressing these problems will definitely improve oil and gas pipeline performance integrity. The type-systems involve the typical pipeline network operational schemes such as the flow rates, pipeline physical component schedule, length and thickness. These schemes are specified clearly to enable the standardized structuring of the models. The semantics in this fluid control mechanism are essentially about the crude and gas parameters relating to molecular weight of measure, relative density, molar and volumetric basis and calorific values. The software craftsmanship has to take all these factors into the engineering framework so as to build a solution that meets the domain specific modeling requirement for solving the stated problems in the solution

Corresponding author; Japheth, B. R. bunakiye.japheth@ndu.edu.ng

Introduction

Oil and gas pipelines like arteries in human life operate 24 hours to transport crude oil from oil fields to destinations where it is turned into several useful products and gas to retail outlets (*Wendy,* 2016). Controlling design dynamics of oil and gas pipelines for effective fluid flow in this context is to achieve a faster and coherent product delivery using domain-specific modeling (DSMForum.org., 2018). It means a domain-specific modeling platform capable of implementing a chronological process for effective fluid flow propagation in a pipeline. A typical system flow involves formulating the flow problem, modeling the geometry and flow domain, establishing the boundary and initial conditions, generating the grid/mesh, establishing the simulation strategy, establishing the input parameters and files performing the simulation monitoring, the simulation for completion, post-processing of the simulation to get the results, making comparisons of the results, repeating the process to examine sensitivities and documentation (Hussain *et al.*, 2016).

The first step of the analysis process is to formulate the flow problem by seeking answers to the following questions: what is the objective of the analysis? What is the easiest way to obtain those objectives? What geometry should be included? What are the operating conditions? What dimensionality (1D, 2D, 3D) of the spatial model is required? What should the flow domain look like? What temporal modeling is appropriate (steady or unsteady)? What is the nature of the viscous flow (inviscid, laminar, and turbulent)? And how should the fluid be modeled (Qin *et al.*, 2016)

The Pipe body and pipeline about which flow is to be analyzed requires modeling, which means the Geometry and the Flow Domain is modeled. This would generally involve modeling the geometry with a CAD software package. The geometry and flow domain would be modeled in such a manner as to provide input for the grid generation (Tolvanen and Steven, 2018). Thus, the modeling would take into account the structure and topology of the grid generation. Since a finite flow domain would be specified, physical conditions are required on the boundaries of the flow domain. The simulation will generally start from an initial solution and uses an iterative method to reach a final flow field solution. The flow domain would be discretized into a grid. The grid generation would involve defining the structure and topology and then generating a grid on that topology. The strategy for performing the simulation would involve determining such things as the use of space-marching or time-marching, the choice of turbulence, chemistry model, and the choice of algorithms (MetaCase, 2017). A program code generally would require an input data file to be created and listing of the values of the input parameters to conform to the desired strategy. Furthermore, the grid file containing the grid and boundary condition information would be required here. Thus, the files for the grid and initial flow solution need to be generated (Meziou *et al.*, 2019). The simulation would be performed with various possible options for interactive or batch processing and distributed processing. As the simulation proceeds, the solution would be monitored to determine if an iterative convergence solution has been obtained. Post-Processing involving extraction of the desired flow properties from the computed flow field is carried out.

This is to ensure that the computed flow properties would be compared to results from analytic, computational, or experimental studies in order to establish the validity of the computed results. The sensitivity of the computed results would be examined to understand the possible differences in the accuracy of results and performance of the computation with respect to dimensionality, flow conditions, initial conditions, marching strategy, algorithms, grid topology and density, turbulence model, chemistry mode, flux model, artificial viscosity, and boundary conditions (Korea, 2019).

Related Work

Wendy, (2016) introduced oil and gas pipelines to arrangements that oil and gas from onshore and offshore oil fields to refineries and to retail outlets nearly on a daily basis. Even though pipelines transport a variety of products such as sewage and water. The most common products transported include products processed from oil and gas. This paper is therefore focusing on devising a software means of how products processed from oil and gas can flow adequately devoid of limited human intervention. Pipelines exist in different sizes and in different forms because of the material used in making the pipes. This pipeline formation may determine the type of goods transported, while some pipelines are built above ground, some are buried underground

Domain-specific modeling (DSM), as adopted in the succeeding considerations is a software engineering methodology for designing and developing systems, such as computer software (DSMForum.org, 2018). It involves systematic use of a domain-specific language to represent the various facets of a system. Domain-specific modeling is simply increased productivity because it raises the level of abstraction and hides programming complexities. This is a whole level of abstraction higher than conventional tools, and makes each symbol worth several lines of code. The application is automatically generated from these high-level specifications with domain-specific code generators, aided where necessary by existing component code.

Ulrich Frank, (2010) outlined a method for designing domain-specific modelling languages. The method description is on a process model, which describes essential steps to be accounted for during the development of a domain specific modelling language. It includes heuristics to develop requirements and meta modelling guidelines that support frequent design decisions. The description of the method is complemented by examples which are mainly taken from the design of the concepts. In recent years, the development of domain-specific modelling languages has gained remarkable attention. This is because domain-specific modelling language usually produced from domain specific modelling incorporates concepts that represent domain-level knowledge. Hence, using domain specific modelling in the oil and gas domain for pipelines design mechanisms will enhance model integrity for flow metrics.

Tolvanen and Kelly, (2018), discussed efforts used to create domain specific modeling languages. Domain-specific modeling languages and generators have been shown to significantly improve the productivity and quality of system and software development. These benefits are typically reported without explaining the size of the initial investment in creating the languages, generators and related tooling. Quite a number of case studies were comparably carried out focusing on the effort to develop a complete modeling solution for a particular domain with the MetaEdit+ tool. Using case study research method, detailed data on the development effort of implementing two realistically-sized domain-specific modeling solutions were achieved. Now comparing with some industry reports, which is complimenting this work, the reports indicate that, for this tool, the investment required to create domain-specific modeling support is modest, and productive. Szopos, (2017) focused on developing new mathematical and computational methods for analyzing biological flows as complex multi-physics and multiscale phenomena. The description of the underlying mechanisms stems from the basic principles of fluid dynamics and is translated into systems of partial or ordinary differential equations. The overall objective was the study of these equations at the continuous and discrete levels, their coupling and the development of a reliable and efficient computational framework to implement various numerical methods to approximate the solutions to these problems. The numerical simulations incorporate realistic geometries, are thoroughly validated against experimental data and target specific biomedical applications. The in-silico approach, namely combining data-driven and physics-driven modeling in a unifying computational framework, has become increasingly popular in the domain of life sciences in the last years and has motivated the development of new mathematical and numerical concepts (Kirch *et al.*, 2020). The fluid dynamics, which is simply utilized in this work is influenced by the possible interaction within the domain concepts. The geometrical representation become very relevant in the perspective that mathematical and computational modeling approaches, based on the fundamental principles and adapted to an oil and gas pipeline framework, can help reduce redundancy in fluid flow activities.

The management of design in the domain of oil and gas pipeline is a whole system activity that help in identifying, handling, and controlling a pipeline design task by coordinating the communication between all component models and dimensions involved in handling the design (Japheth, and Asagba, 2015). This is achieved by allocating and managing resources, and by providing access to relevant design-related information to domain experts Their submission contained general informal requirements for a pipeline design management system (PDMS), a feature model for the PDMS product line, and a domain model for the PDMS, as well as an informal physical pipeline model of the PDMS. Domain specific modeling researchers who want to demonstrate the power of their technique can hence apply the approach in other related areas at the most appropriate level of abstraction in the domain of pipeline engineering.

Amine *et al.*, (2019), presented a reduced–order thermal fluid dynamic model for gas/liquid two–phase flow in pipelines. Specifically, a two–phase–flow thermal model is coupled with a two–phase–flow hydraulics model to estimate the gas and liquid properties at each pressure and temperature condition. The proposed thermal model estimates the heat–transfer coefficient for different flow patterns observed in two–phase flow. For distributed flows, where the two phases are well–mixed, a weight–based averaging is used to estimate the equivalent fluid thermal properties and the overall heat–transfer coefficient. Conversely, for segregated flows, where the two phases are separated by a distinct interface, the overall heat–transfer coefficient is dependent on the liquid holdup and pressure drop estimated by the fluid model. Intermittent flows are considered as a combination of distributed and segregated flow. The integrated model is developed by dividing the pipeline into segments. Equivalent fluid properties are identified for each segment to schedule the coefficients of a modal approximation of the transient single–phase–flow pipeline–distributed–parameter model to obtain dynamic pressure and flow rate, which are used to estimate the transient temperature response. The resulting model enables a computationally efficient estimation of the pipeline–mixture pressure, temperature, two–phase–flow pattern, and liquid holdup. Such a model has utility for flow–assurance studies and real–time flow–condition monitoring (Szopos, 2015). However, incorporating a domain specific reusable software layer can influence the design dynamics for better flow assurance.

Domain Specific Modelling Utility

Solid modeling (Japheth, and Ogude, 2019), is the process of building up of solid models (objects) from primitives. Primitives are geometric shapes found in common computer aided design (CAD) systems such as AutoCAD. Possible operations found in common CADs for solid modeling include extrusion, revolution, sweep and Boolean operations such as union, subtraction and intersection. Though it is possible to display in various ways and also to determine some material properties, solid models as described are referred to as dump solids because it is difficult to change the geometry, their features contain only geometric information, the dimensions are static, and i.e., they do not drive the geometry, and the order of operation of the model is unavailable to the user (Japheth and Acheme, 2017). Parametric modeling approach, recently associated with common interactive CAD systems such as AutoCAD however addresses all of the shortcomings of dump solids. Parametric modeling has become a basis for integration of differing possibilities of engineering designs modeling in a homogenous platform and has provided the ability to reflect the way in which modern manufacturing companies develop their products. With parametric modeling current Computer-aided Design systems are tools and software packages range from 2D vector-based drafting systems to 3D solid and surface modelers (Qin *et al.*, 2016). They are used for detailed engineering of 3D models of physical components, and also used throughout the engineering process from conceptual designs and layout of products, through strength and dynamic analysis of assemblies to definition of manufacturing methods of components. Parametric modelling employs parametric dimensions and geometric constraints to define features, and to create relationships between these features in order to create intelligent models. Though this flexibility gives the user enough room to explore design

alternatives, it is still difficult for domain-specific concepts to be expressed easily during design (MetaCase, 2017).
Even in conventional modeling tools, the instructions used to build a model from scratch, each time use same parameters or experimenting with different ones. The parameters can be numeric values, relationships, and can even include graphic parameters already existing in the model (e.g., angular pipes, etc.). The basic challenge is the issue of interaction between models, interactions in the way of concepts devoid of possible parametric constraints within a CAD system. Interactions that can produce other complete models with noticeable properties relative to a given set of concerns in relevant domains that captures accurately and concisely all of its interpretation and design intent for specific problems and solutions. This has not been achieved with current conventional modelling systems. Rather domain specific modelling, which invariably metamorphosis into a domain specific language (DSL) can have the sufficient linguistic power to handle domain and platform complexities and has moved speedily with domain technologies, capable of handling the design dynamics of oil and gas pipeline for productivity (Chabannes *et al.*, 2019).

Domain Specific Languages

Domain specific languages (DSLs) are mini or special purpose languages that have been designed and tailored for specific application domains. A domain specific language is a language that is provided by a programming library or Application Programming Interface (API) for an existing general-purpose language. This means in combination with an application library, any GPL can act as a DSL. The library's Application Programmers Interface constitutes a domain-specific vocabulary of class, method, and function names that becomes available by object creation and method invocation to any GPL program using the library (Haupt, 2020). The reasons for developing DSLs are that they offer established domain specific notations beyond the limited user-definable operator notation offered by GPLs. Appropriate domain-specific constructs and abstractions cannot always be mapped in a straightforward way by GPL to functions or objects that can be put in a library. A GPL in combination with an application library can only express these constructs indirectly or in an awkward way. Traversals and error handling are typical examples. They also offer possibilities for analysis, verification, optimization, parallelization, and transformation in terms of DSL constructs that would be much harder or unfeasible if a GPL had been used because the GPL source code patterns involved are too complex or not well defined (Hussain *et al.*, 2017).

Addressing the Research Problem

Presented here is the general framework of how the central concepts address the research problem. Central concepts of the research include the provision of efficient, flexible, and extensible reusable software with familiar notations capable of handling the stakeholder viewpoint at the workplace regarding pipeline activities at both the downstream and upstream sectors of the oil and gas industry in Nigeria. A software solution with secure and scalable domain model selections for onward fabrication that can enhance the entire oil and gas industry business life cycle in Nigeria, standard and consistent valuation data throughout design interests and divisions in pipeline projects that can account for the economic success in the oil and gas industry in Nigeria, the operational analysis of artifact orientation options, and diversifications that can cater for real time stakeholder variability management (*Wendy*, 2016). Now addressing the research problem; the trade-off of switching to using domain specific modelling system is reduction in design time and cost of operations. The simplicity in usability of domain specific modelling systems is clearly captured in the vocabulary of components and then translated into a modelling platform (i.e., the efficient, flexible, and extensible reusable software with familiar notations). This software solution with secure and scalable domain model selection capacity can still solve the problem in optimal cases, the clear benefits are that; the metamodel, usually, is a repository of concepts from the oil and gas pipeline domain. These models are constructed using the concepts and not the concepts of a given programming language, therefore, the semantic gap between design intent and the expression of this intent in several lines of codes is bridged for optimal performance in controlling a typical scenario (Carichino *et al.*, 2017).

The Language Metamodel

Domain specific modeling approach is declarative, usually expresses what the program should accomplish by hiding from the user the complexities of how to solve the problem in terms of sequences of actions to be taken. Policies are specified at a higher level of abstraction using models and are separated from the mechanisms used to enforce the policies. This makes it possible for virtual prototyping to be the way to go as the need to ensure designs perform efficiently in the field before physical models are setup (Ulrich, 2010). The key features to consider in order to ensure efficient design performance include heat balance. The heat balance equation may now be written as:

$$\left\{ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \frac{\partial}{\partial z} \right\} [k] \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial x} \end{bmatrix} + Q = c\rho \frac{\partial T}{\partial t} \quad \ldots\ldots 1$$

When cylindrical coordinate r, ϕ, and z are used, as for a body of revolution as in pipe section, the heat balance equation is derived as:

$$\left\{ \frac{\partial}{\partial r} \frac{\partial}{r\infty} \frac{\partial}{\partial z} \right\} \begin{bmatrix} k \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial r} \\ \frac{1\partial T}{r_{\partial \emptyset}} \\ \frac{\partial T}{\partial z} \end{bmatrix} + Q = c\rho \frac{\partial T}{\partial t} \quad \ldots\ldots 2$$

The material is usually assumed to be thermally orthotropic with one axis of orthotropic coinciding with the circumferential Φ direction (Bertoluzza *et al.*, 2017). Then [k] is of the form with $k_{x'}$ and $k_{y'}$ replaced by the equivalent conductivities in the r-z plane while kz′ is replaced by the conductivity k_{Φ}. Boundary conditions that may be imposed are the following:

(a) temperature T* may be prescribed on portion S_T of the boundary S
(b) heat flux q* may be prescribed on portion S_T of the boundary S. Since heat flow occurs normal to the boundary, the prescribed heat flux is given by:

$$\vec{q} \cdot \vec{n} = q_x n_x + q_y n_y + q_z n_z = - \{n_x\ n_y\ n_z\} \begin{bmatrix} k \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial x} \end{bmatrix} = q^x \quad \ldots\ldots 3$$

With $\vec{n}$ the vector normal to the surface sq and n_x, n_y, and n_z components if the surface is insulated.

$$q^* = 0 \quad \ldots\ldots 4$$

(c) convective heat transfer conditions may be prescribed on portion S_C of S.
Then

$$\vec{q} \cdot \vec{n} = h\,(T - T\infty) \quad \ldots\ldots 5$$

with h the convective heat transfer or film coefficient and T_∞ the fluid temperature. In addition, the temperature distribution within the body must be prescribed at time t=0.

Then $$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} T_n \quad \ldots\ldots 6$$

With $$B = \begin{bmatrix} \frac{\partial \Delta}{\partial x} \\ \frac{\partial \Delta}{\partial y} \\ \frac{\partial \Delta}{\partial z} \end{bmatrix} \quad \ldots\ldots 7$$

The portion of π_T contributed by each element is then given by T

$$\begin{bmatrix} \Delta\pi_T \end{bmatrix} = \{ T_n \}^T \ ½ \begin{bmatrix} k_T \end{bmatrix} + \begin{bmatrix} h_T \end{bmatrix} \{ T_n \} - \{ rQ \} + \{ r_q \} - C_T \ \frac{\partial T_x}{\partial t} - \{ r_\infty \} \quad \ldots\ldots 8$$

With $\left[k_T\right] = \iiint_{\Delta v} \left[B\right]^T \left[k\right]\left[B\right] dv$ ……9

$h_T = \iint_{\Delta sc} h \left[D\right]^T \left[D\right] ds$ ……10

$\{ rQ \} = \iiint_{\Delta v} Q \left[D\right]^T dv$ ……11

$\left[rq\right] = \iint_{\Delta sq} q^* \left[D\right]^T ds$ ……12

$C_T = \iiint_{\Delta V} ec \left[D\right]^T \left[D\right] dv$ …… 13

$r_\infty = \iint_{\Delta sc} hT_\infty \left[D\right]^T ds$ ……14

Were
ΔV = volume of element
ΔSc = portion of the boundary Sc which is a boundary of the element
Δsq = portion of the boundary Sq which is a boundary of the element. For interior elements, these are equal to zero. On the remaining portions of the boundary S_T, this temperature at nodal points is given by T^*. Assembly of the variational functional π_T yields an expression of the form

$\pi_T = \{T\}^T \{ ½ \{ \left[k_T\right] + \left[H_T\right] - \{ R_Q \} + \{ R_q \} + \left[C_T\right] \frac{\partial T}{\partial t} - R_\infty \}$ …… 15

Where {T} is the matrix of the nodal temperature arranged sequentially and the remaining matrices are the assembled versions of those defined by eq. (8) to eq. (Chabannes *et al.*, 2017). Minimization with respect to the nodal quantities yields the following set of equations:

$C_T \left[\frac{\partial T}{\partial t}\right] + \left[k\right] \{T\} = \{ R. \}$……16

With

$\left[K\right] = \left[k_T\right] + \left[H_T\right]$ ……17

And $\{R\} = \{ R_Q \} + \{ R_\infty \} + \{ R_q \}$……18

Those equations of the above set which represent minimization with respect to nodal temperatures on S_T should be deleted. In the remaining equations those terms of ∂T/∂t referring to temperatures on S T vanish while the corresponding terms in T are set equal to the prescribed temperature (Chabannes *et al.*, 2019). This approach can be applied easily if the prescribed temperature is zero since the node at which that temperature is prescribed need not be included in the numbering system. For a non-zero prescribed nodal temperature T*, another approach is to add a large value of conductivity K_n to the corresponding diagonal coefficient of [K] and to replace the corresponding coefficient in {R} by $K_nT_n^*$ · This method effectively forces the nodal temperature to be equal to the prescribed value for sufficiently large K_n.
The analysis of steady-state heat conduction problems involves the solution of a set of simultaneous equations given by Eqs. (16) with each term of ∂T/∂t set equal to zero and with Q and q * assumed to be independent of time. This may be accomplished, as for static stress · analysis, by Gaussian elimination. The analysis of transient heat conduction, however, involves the solution of Eqs. (16) as a set of simultaneous first order linear differential equations in time subject to certain initial conditions (Yakovlev *et al.*, 2019). These equations may be solved by mode superposition, as for dynamic stress problems, by determining the eigenvalues and eigenvectors of the equation

$k - \lambda C_T \{T\} = \{0\}$ ……19

A more usual approach is to numerically integrate the differential equations. One such method uses the assumption that temperatures at time t and t+Δt are related by

$$\{T\}_{t+\Delta t} = \{T\}_t + \{ (1-\beta) \{ \frac{\partial T}{\partial t} \} + \beta \{ \frac{\partial T}{\partial t} \}_{t+\Delta t} \} \ldots\ldots 20$$

By writing Eqs. (29) for time t and t+Δt, the derivatives of temperature can be eliminated and a set of equations for temperature at time t+Δt can be obtained as

$$(\frac{1}{\Delta t} C_T + \beta K) \{T\}_{t+\Delta t} = \{ \frac{1}{\Delta t} C_T + (1-\beta) K\}(T)_t + (1-\beta) \{R\}_t + \beta \{R\}_{t+\Delta t} \ldots\ldots 21$$

The above conservative representation of flow variables are clearly partial differential equations and to be able to solve these equations using digital computers there is need to use the process of Numerical Discretization. In this discretization process, each term in the partial differential equation must be translated into a numerical analogue that the computer can be programmed to solve. Thus, the problem is reduced to the repeated solution of creating a metamodel of the domain concepts for effective design activity on the computer (Chabannes *et al., 2*019).

Conclusion and Future Work

Using domain specific modelling can produce a system capable of handling design dynamics of oil and gas pipeline for productivity. It is possible with a system, comprising a domain server, comprising a pipeline-built units' memory, configured to store a set of attributes comprising at least one attribute relating to a range of components in the lifecycle of pipeline design operation. Each of the range of components is representative of a type of operation for a pipeline-built unit, the range of components including at least two involving a drilling operation phase, a transmission operation phase and a production completion phase for the entire pipeline-built units. A domain server, comprising a processor configured to generate an orientation for performance operation of the first instance of the pipeline-built units based on the evaluation of associated models of the first pipeline instance for at least a component of the range of components from information of one or more other data relating the pipeline context model data. A more permanent computer readable translation engine merged to the domain server; and managing information of one or more other built units containing instructions for a process for the domain server; the instructions comprising at least one attribute relating to a range of components in the lifecycle of the entire pipeline design operation. A method for the domain server, where the set of attributes are identified as attributes associated to at least one of reducing design time, express and achieve design intents, and increasing production output. The expected results of the system would be: Pipeline engineers need not consult the services of professionals at the design stage. It will take away time consuming efforts, and huge cost incurred during design in oil and gas pipeline engineering project. It will help to integrate stakeholders determining factors simply on one mechanism to generate artefacts for the design scenarios easily. Help Stakeholders make investment decisions with greater confidence where changes are applied to extensible parameters including attributes, values, dimensional standards, operational standards, and regulations. Implement software in the oil and gas industry in Nigeria in a multi-user and transmission pipeline physical asset-wide level under controlled environment where data access is only familiar and limited to stakeholders' views and changes. Introduce more structured design analysis and processes for more consistent economic industry evaluations

References

Bertoluzza,S. V. Chabannes, C. Prud'homme, and M. Szopos. "Boundary conditions involving pressure for the Stokes problem and applications in computational hemodynamics". In: Computer Methods in Applied Mechanics and Engineering 322 (2017), pp. 58–80.

Chabannes V., M. Ismail, C. Prud'homme, and M. Denis Jose Schiozer, Antonio Alberto de Souza dos Santos, Susana Margarida de Graca Santos and Joao Carlos von Hohendorff Filho (2019), Model-based Decision Analysis Applied to Petroleum Field Development and Management', Oil Gas Sci. Technol., Vol. 74, Article 46; https://doi.org/10.2516/ogst/2019019

Carichino L., G. Guidoboni, and M. Szopos. "Energy-based operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows: the Stokes case". In: 7th International Conference on Computational Methods for Coupled Problems in Science and Engineering, Rhodes, Greece (2017), full paper in revision (2017).

Chabannes,V. C. Prud'homme, M. Szopos, and R. Tarabay. "High order finite element simulations for fluid dynamics validated by experimental data from the FDA benchmark nozzle model". In: 5th International Conference on Computational and Mathematical Biomedical Engineering (CMBE 2017) Proceedings. 2017, pp. 1150–1153.

DSMForum.org. 2018. http://dsmforum.org/cases.html

Hussain, A.; Vega, J.C.; Hassane, M.A.S.; Yusaf, S.A.; Abdul-Halim, A.A. Enhancing Smart

Completion Capabilities by Integration with Digital Oil Field Real Time Monitoring System in a Green Field of ADMA-OPCO. In Proceedings of the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, 7–10 November 2016.

Haupt, M. "Data is the New Oil"—A Ludicrous Proposition. Available online: https://medium.com/project-2030/data-is-the-new-oil-a-ludicrous-proposition-1d91bba4f294 (accessed on 7 July 2020).

Japheth, Bunakiye R and Ogude U. Cyril; (2019), A Domain Specific Modeling Language Semantic Model for Artefact Orientation: International Journal of Cognitive and Language Sciences Vol: 13, No: 2, 2019, Open Science Index, waset.org/abstracts/99162.

Japheth Bunakiye Richard, Asagba Oghenekaro Prince. Oil and Gas Pipeline Design Management System: A Case Study for Domain-Specific Modeling. American Journal of Software Engineering and Applications. Vol.4, No. 5, 2015, pp. 92-98. doi: 10.11648/j.ajsea.20150405.13.

Japheth, Bunakiye R. and Acheme I. David (2017), 'Defining a DSL for Transmission Pipeline Systems Meta-Modeling', Future Technologies Conference (FTC), 2017, Vancouver, Canada

Kirch, A., Razmara, N., Mamani, V.F.S. *et al.* (2020), Multiscale Molecular Modeling Applied to the Upstream Oil & Gas Industry Challenges. *Polytechnica* (2020). https://doi.org/10.1007/s41050-019-00019-wMetaCase. 2017. MetaEdit+ 5.5 User's Guide. http://www.metacase.com/support/ 55/manuals/

Korea Institute of Geoscience and Mineral Resources. Smart Evaluation Solution for Oil and Gas Field Based on Integration of Static and Dynamic Data; KIGAM: Daejeon, Korea, 2019.

Meziou, Amine , Khan, Zurwa , Wassar, Taoufik , Franchek, Matthew A., Tafreshi, Reza , and Karolos Grigoriadis. "Dynamic Modeling of Two-Phase Gas/Liquid Flow in Pipelines." *SPE J.* 24 (2019): 2239–2263. doi: https://doi.org/10.2118/194213-PA.

Marcela Szopos. Mathematical modeling, analysis and simulations for fluid mechanics and their relevance to in silico medicine. Analysis of PDEs [math.AP]. Université de Strasbourg, IRMA UMR 7501, 2017. fftel-01646867f

Tolvanen Juha-Pekka and Steven Kelly. 2018. Effort Used to Create DomainSpecific Modeling Languages. In ACM/IEEE 21th International Conference on Model Driven Engineering Languages and Systems (MODELS '18), October 14–19, 2018, Copenhagen, Denmark. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3239372.3239410.

Qin, H.; Han, Z. Stochastic Resource Allocation for Well Control With Digital Oil Field Infrastructure. IEEE Syst. J. 2016, 12, 1295–1306. [CrossRef]

Szopos. "Hemodynamic simulations in the cerebral venous network: A study on the influence of different modeling assumptions". In: Journal of Coupled Systems and Multiscale Dynamics 3.1 (2015), pp. 23–37.

Ulrich Frank ICB-Research Report No. 42 December 2010 Outline of a Method for Designing Domain-Specific Modelling Languages https://www.econstor.eu/obitstream/10419/58163/1/716089785.pdf

Yakovlev, V.V., Kalugin, Y.I., Kalugin, A.Y. *et al.* (2019)., Mathematical modelling of enhanced gas production. *J Petrol Explor Prod Technol* **9,** 561–571 (2019). https://doi.org/10.1007/s13202-018-0495-7

Wendy, F. (2016, JUNE 14). An introduction to oil and gas pipelines. https://www.fractracker.org/2016/06/introduction-oil-gas-pipelines/