Bunakiye R. Japheth

# Application of Model Driven Valuation Methodology for Maximum Throughput in Oil and Gas Pipeline Engineering Operations

bunakiye.japheth@ndu.edu.ng
Bunakiye R. Japheth
Department of Computer Science,
Niger Delta University Wilberforce Island Bayelsa State, Nigeria.



The link to this publication is **https://ajoeer.org.ng/otn/ajoeer/2022/qtr-1/04.pdf**

**Dr. B R Japheth** obtained Bachelor of Science (Mathematics) from Federal University of Technology Minna, Niger State in 2002 and Master of Science (MSc Computer Science) in 2007 and Doctor of Philosopphy (PhD Computer Science) in 2016 from University of PortHarcourt, Rivers State, Nigeria. He is currently a Senior Lecturer in the Department of Computaer Sciences, Niger Delta University, Bayelsa State, Nigeria. He is a member of Computer Professionals Council of Nigeria (CPN) since 2009 and IAENG Association of Engineers and Computer Scientist since 2018. He has published more than 25 research papers in reputable international journals including SciencePG(Software Eng.) and conferences including IEEE (SAI), and are also available online. His main research work focuses on Requirements Engineering, Domain Specific Modelling, Theory of Computation, Formal Specification and Verification,.and Cybersecurity. He has 12 years of teaching experience and 10 years of Research Experience.

# Application of Model Driven Valuation Methodology for Maximum Throughput in Oil and Gas Pipeline Engineering Operations

Bunakiye R. Japheth
Department of Computer Science,
Niger Delta University Wilberforce Island Bayelsa State, Nigeria.

*Email: bunakiye.japheth@ndu.edu.ng*

## ABSTRACT

The focus of this paper is on economic success for engineers and related experts whose primary line of business is tied to the field of pipeline mechanisms in the oil and gas industry. General purpose programming languages and computer aided design systems cannot express oil and gas pipeline engineering domain concepts appropriately, which means that domain experts in the oil and gas industry find it cumbersome to operate at optimal using conventional software in their line of business. This paper, therefore, presented a model driven valuation technique that enable the engineer simply input values corresponding to familiar notations such as pipe diameter, fittings, flow metrics etc.) on an interface to get done the kind of design, simulation artifact and other pipeline configurations without having to use any general-purpose programming system. The policies and mechanics of the valuation coupled software system production process incorporated inherent properties from MATLAB application programmer interface (API) to integrate the domain characteristics and concepts. The result became a new layer of reusable software capable of processing these domain characteristics and concepts to produce desired artefacts. A collection that supports further layers of accessible software connected through interfaces. This system is recommended to industries with a principal business interest in the oil and gas industry with the hope that crafting pipeline configurations for effective oil and gas supply within the value chain can be achieved with ease.

***Keywords***: pipeline project, consistent valuation data, design interests, model driven design, domain driven design, design scenarios, economic success.

## 1 Introduction

Pipeline systems are like arteries and veins; they are an essential and integral part of our modern civilization just as arteries and veins are essential to the human body. In a modern city they transport water from the sources of water supply to the points of distribution. Similarly, pipelines carry crude oil or gas from wells to tank farms for storage or to refineries for processing [18]. This simply means that pipeline systems are useful both in the upstream, midstream and downstream operations in the oil and gas industry. In the upstream sector, pipelines are very critical assets in production activities, which include onshore and offshore drilling. In both onshore and offshore drilling, smart technologies are used to enhance the drilling efficiency and well performance by providing real-time information and trends. For example, in onshore drilling facilities, the wells are grouped together in a field, ranging from a half-acre per well for heavy crude oil to 80 acres per well for natural gas. The group of wells are usually connected by carbon steel tubes which sends the oil and gas to a production and processing facility for further treatmemt. In offshore drilling, single platform that is either fixed or floating are used abd pipelines are used to transport the oil and gas to a production and processing facility for further treatmemt [22]. The midstream sector covers transportation, storage, and trading of crude oil, natural gas, and refined products. oil and gas in its unrefined state, is primarily transported by pipelines. Once the oil has been extracted and separated from natural gas, pipelines transport the products to another carrier or directly to a refinery. Petroleum products then travel from the refinery to market by tanker, truck, railroad car, or more network of pipelines. The downstream sector covers refining and marketing. Refining invoves transforming crude oil into petroleum products used for a variety of purposes such as heating homes, fueling vehicles and making petrochemical plastics. Marketing is the wholesale and retail distribution of refined petroleum products to business, industry, government, and public consumers [20]. Generally, all these products flow to far markets and other destinations through pipelnes, which means, adopting a methodology that can technically optimize pipeline engineering operation for maximum productivity in all streams in the oil and gas industry is now needed. The design, construction, operation, and maintenance of various pipeline systems involve understanding of pipeline fundamentals, materials, generic and specific design co3nsiderations, fabrication and installation, examinations, and testing and inspection requirements, in addition to the local, state and federal regulations [16]. Pipeline systems include pipe, flanges, fittings, bolting, gaskets, valves, and the pressure containing portions of other piping components. It also includes pipe hangers and supports and other items necessary to prevent over pressurization and overstressing of the pressure-containing components [14]. Engineers responsible for the preparation of design documents must, from time to time, review the current codes and standards in order to comply with and take advantage of the changes in the industry which are expected to continue as computerized drafting and preparation of text and record keeping improvement strategies. Before the detailed design is begun, it is usually advantageous for the project to prepare a comprehensive set of documents defining the system design criteria. This criterion may be part of overall project design criteria or may be a separate document prepared solely for the piping design [11]. In either case, the design criteria should reiterate the design requirements delineated in the contract specification and should define the applicable codes and standards, environmental conditions, design parameters, and other pertinent design bases that will govern the work. This paper is therefore hinging on recent model driven software development valuation methodology to ensure that pipeline engineering processes for oil and gas transportation and the associated design criteria be updated as the design progresses to reflect any change in the design basis [21]. So that maximum throughput that connects all the stream in the oil and gas industry productivity value chain is accomplished.

## 2. Related Work

The many organizations engaged in pipeline system design do perform system descriptions (SDs) based on their needs. Systems whose operation may have public safety implications (e.g., nuclear power plant systems and chemical plant systems) may require very detailed system descriptions [19]. They further defined that the purpose of pipeline system description is to set forth, specifically in writing, the functions, intent of the design, and major features of the system [17]. Some common features include system design bases, operating modes (start-up, normal, shutdown, emergency, or as appropriate), performance ratings of major equipment, and control concepts. The design basis stated in the SD is used to develop the system flow diagram (which shows the features necessary to accomplish the design basis) and the pipeline diagram (which shows the basic controls, interlocks, pipes, valves, vessels, pumps, and miscellaneous equipment comprising the system). Therefore, the system description is important in the development of the documents used in procurement, manufacturing, fabrication, and erection [20]. The importance of the adoption of isometric or orthographic pipeline sketches methodology in pipeline systems design for maximum production in the oil and gas industry was resounded as determined by project requirements [15]. Portions of structural, and control information may be included in these sketches in which case they are called composite drawings [13]. These drawings form the basis for the working physical drawings such as the system isometrics. The sketches and composites are not used for construction or manufacture; they represent the pre-computer version of the present database in a computer-aided design (CAD) system, or one of its several derivative systems, which can provide the design study more efficiently than manual methods in many instances today. Computer-aided design and drafting (CADD) software packages are available from a number of commercial sources [6]. With the development of three-dimensional computer-aided design (3DCAD) software, the designer can check for interference and can generate different views. Once the orthographic drawings are completed, they may be issued for piping fabrication and construction. However, for complex pipeline systems, it is common practice to develop separate piping isometric drawings for each pipe run [6]. For pipe stress analysis, fabrication, and installation, the piping isometric drawings are easier to use than the orthographic drawings because all the information on the isometric drawing pertains to the pipeline of interest without cluttering with extraneous information [18].

Domain Driven Design (DDD), which is a design pattern in model driven software development focuses software development on domains and domain logic [1]. The principle of DDD widely acknowledges that domain specific modeling is central to software design. Through domain modeling, the developments of projects facing complex domains are expressed in rich functionality and translate that functionality into software implementation that truly serves the needs of its users [3]. These principles are applied here because the development and implementation of a pipeline project involves so much design efforts even with the adoption of CAD systems. CAD systems are characterised with numerous compliance issues even after design is completed and the project kicked off. With DDD, the pipeline physical components value and attributes (i.e., the model) is the heart of the design, and the methodology of manipulating this model to ease the project execution for users in the model driven development paradigm [8]. Keeping these CAD drawings as the model and presenting a domain model of them in a metamodel of relationships will result in a modeling language. The modeling language, which is specific to the domain of pipeline engineering, will be capable of allowing the pipeline engineer to easily get the specifications correct on time and at very minimal costs instead of always returning to conformal forms and related issues [4].

Tool integration became a common requirement in modelling software development processes. The idea is that performing the domain functionality to aid design decisions especially for pipeline systems in the oil and gas industry is a common requirement for valuation methodology adoption [2]. The same way this paper is utilizing valuation method to enhance oil and gas pipeline design, there has been a surge of interest in applying model driven engineering (MDE) technologies tool integration. Instead of defining each dynamism separately on a different framework before integrating to achieve desired model transformations, the application of model driven engineering (MDE) technologies will further initiate model transformations and syntactic and semantic interoperability via tool utilization [[5].

## 3. Layered Approach to Model Driven Development

Conventionally [2], model driven software development involves as first steps requirements engineering and domain analysis. Requirement's engineering determines defining the domain, defining and designing the language constructs that accurately captures the domain semantics, develop the domain instances using the new language infrastructure, and building the software tools to support the language [4]. It also involves requirements analysis and documentation, and evaluation baselines for continuous assessment of the metrics of success and improvement of the software. For the language formalism to capture the domain concepts clearly, the model will be specified and incorporated in the domain specific modelling infrastructure as DSM definition and DSM use (See fig. 1).

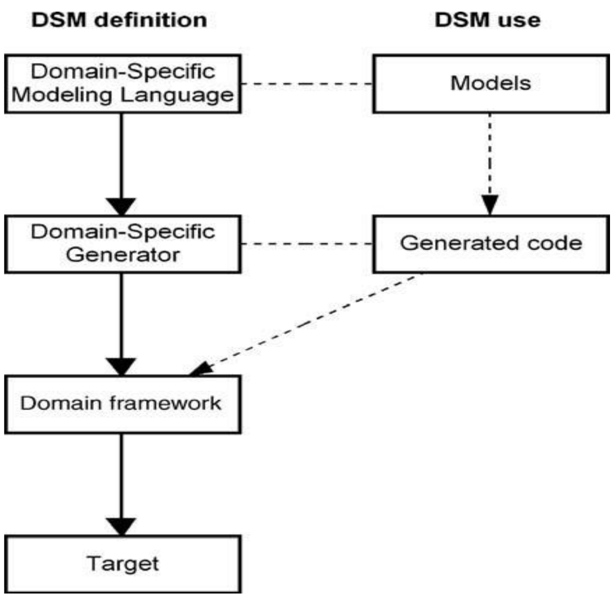


**Figure 1: DSM Definition and DSM Use (source:** Domain Driven Design Martin Fowler, 2020**)**

The processes result into several benefits such as improved user satisfaction, quality in concrete syntax, and complexity hiding [23]. To get these benefits, there has to be some specifications on how the computational mechanism from the domain models will work in running the systems. In the related tasks of DSM definition and use; the domain-specific language provides the abstraction mechanism to deal with complexity in the given domain. Generally, at this stage the major domain concepts map to modeling language objects, thus, the language defined as a metamodel with related notation and tool support; allows developers to perceive themselves as working directly with domain concepts [23]. The code generator specifies how information is extracted from the models and transformed into code. In the simplest cases, each modeling symbol produces certain fixed code, including the values entered into the symbol as arguments. This code will be linked with the domain framework and compiled to a finished executable system. While creating a working solution the objective is that after code generation through the valuation additions, additional manual effort to modify or extend the generated code is not needed. The generated code is thus simply an artefact to the finished product [2]. A domain framework provides the interface between the generated code and the underlying platform. It can directly call the platform components, whose existing services are enough to make the generated code simpler. This domain framework can range in size from components down to libraries, which provide predefined building blocks [3]. The DSM architecture also allows progression. Any of the elements can be changed if needed. This flexibility makes the DSM approach different from Computer Aided Software Engineering (CASE) and fourth generation languages (4GLs), which fix at least one of the architectural elements [7]. One notable way is making sure the domain framework is not visible to the users, in a similar manner as BIOS code or primitives called by the running application are not visible to programmers in a general-purpose programming language (GPL). Another progression is in the aspect of the software that specifically solve problems from the domain, which see the elements of the domain model as a system. These aspects of progression are possible because the generated code is not executed alone but rather together with additional code in some target environment. This target (e.g., .NET, MATLAB), comes with platform code i.e., code that is formalized into a metamodel, where all the models' describing applications or features are instantiated from this metamodel. Thus, these models can't express anything else other than what the language allows. This language instantiation ensures that application developers follow the concepts and rules of the domain models [4].

### 3.1 The Design Considerations

The focus in this section is mainly on the architectural formality as shown in figure 2 that encompases flexibility in at least one of the domain specific elements [4]. The flexibility feature clearly is an embodiment of a collection of modelling primitives and the rules connecting them and in addition, the domain descriptions recursively defined for pre-processing through the system functions.

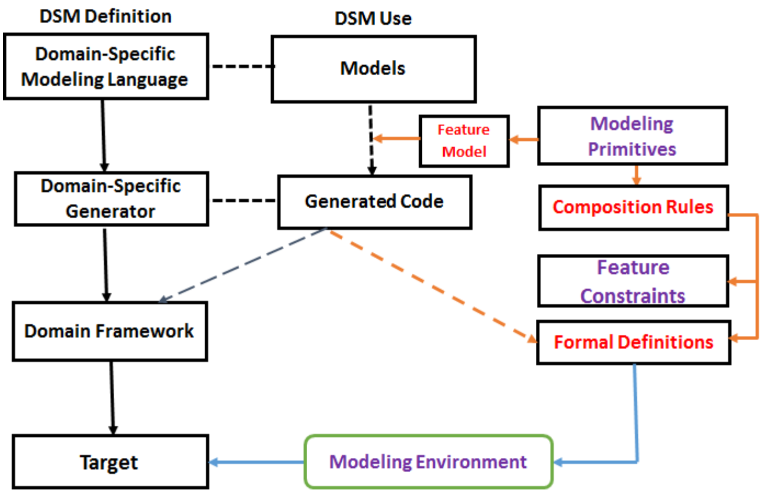


**Figure 2: Model Driven Valuation Methodology Components**

The aspect of the system that is responsible for processing involves the Decision Repository model containing the data model, the pipeline system metadata and the Translation Engine. The decision repository is the store house of the specifications of the semantic domain and its operations; in which case, the pipeline is the root concept, meaning that it is the target result of all the underlying interdependencies of the components interactions [5]. The data model therefore, describes the pipeline instances and objects manipulated by the in-memory object model with the state of events incorporated for actual evolution of virtual pipeline routes. The data model design as shown in figure 3 precedes the creation of the relevant tables to match the domain model metrics. The user queries are responded to by the system through the domain model actions in the syntax-oriented translation function related to the user interface (UI.). Accessibility is one to many relations where pipeline component (PC) types can have as many fittings types parameter values a potential user can input. Then the data binding actions from the executable platform correctly integrates the subsystems to open up control points for integrating into the translator [7].

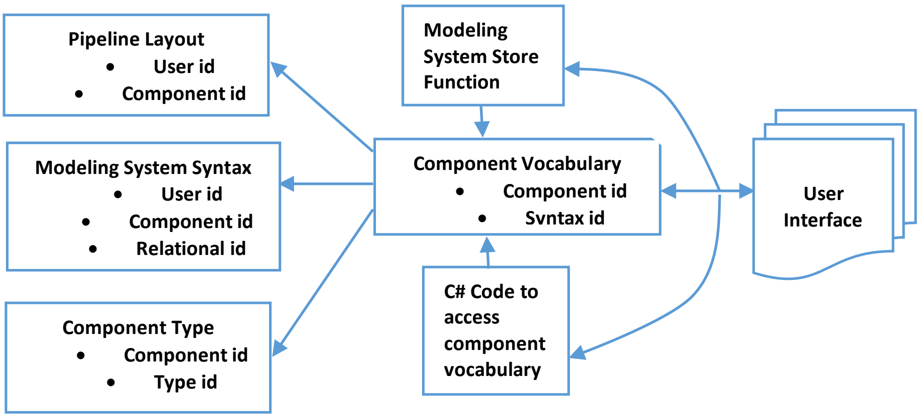


Figure 3 The System Data Model

The language domain model contains logical underlying engineering principles, that leads to identifying the classes and attributes in the data model. The identified elements illustrate how they are linked to produce a total life cycle approach to pipelines system design and operation. The metamodel aspect of the underlying engineering principles in the domain model logic are as shown in figure 4 [4]. This is the metadata of the relationships of the instances and their interdependencies to producing a total pipeline system design life cycle.

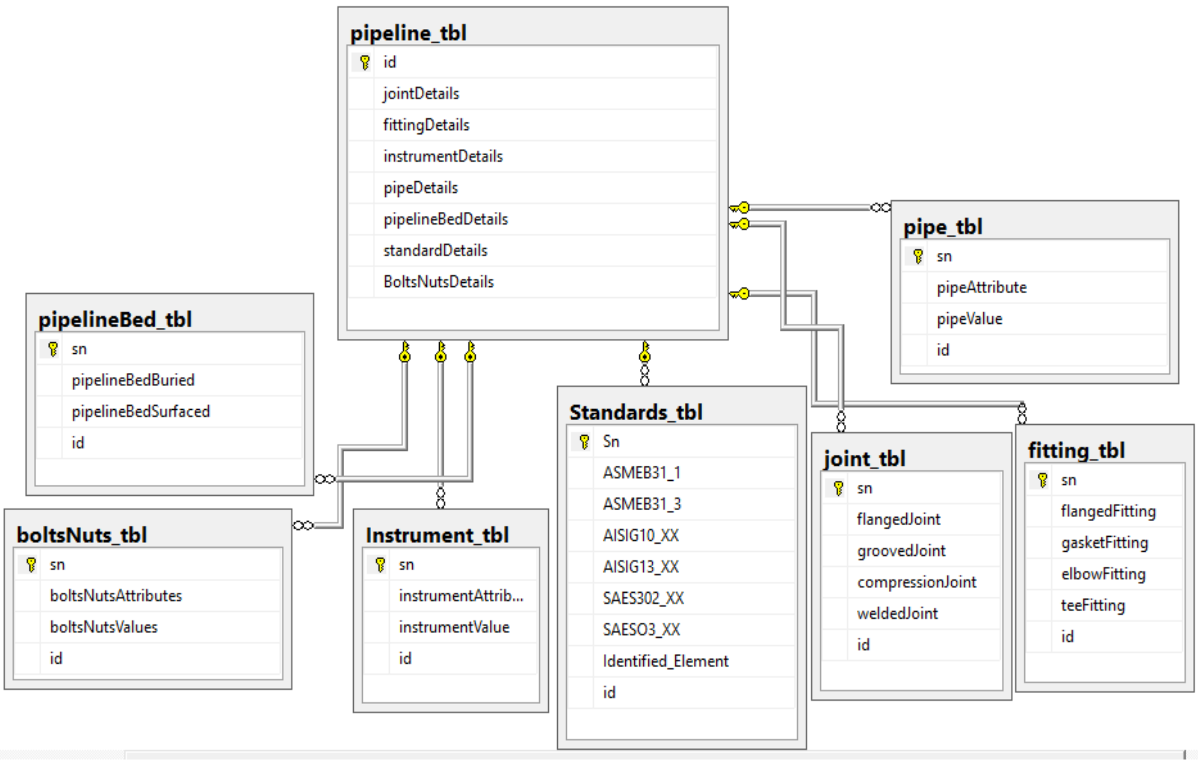

Figure 4. Metamodel of Pipeline Data

The model driven procedures applied in this context in order to achieve maximum throughput in the oil and gas pipeline engineering operations are simply data binding actions from the executable platform that correctly integrates the subsystems [5]. These integration-oriented actions are valuation functions, which are specified as a set of corresponding BNF rules for the domain. The essence and the software development value addition are that the data binding actions, which correctly integrates the subsystems will open up control points for integrating into the translation engine as shown in figure 5.

$$Let\ Component\ Functionality\ be\ (fc)$$
$$For\ a\ given\ design\ operation\ (C_e):$$
$$C = f(R)\ \ldots$$
$$where$$
$$fc: C_1d_1t_1 \times C_2d_2t_2 \times C_3d_3t_3 \times \cdots\cdots \times C_e \longrightarrow R$$
$$fc: Cp_1f_1d_1t_1 \times Cp_1Cj_1d_1t_1 \cdots\cdots \times C_e \longrightarrow R$$
$$fc: Cp_2f_2d_2t_2 \times Cp_2j_2d_2t_2 \times \cdots\cdots \times C_e \longrightarrow R$$
$$fc: Cp_3f_3d_3t_3 \times Cp_3j_3d_3t_3 \times \cdots\cdots \times C_e \longrightarrow R$$
$$\vdots$$
$$\vdots$$
$$fc: R_1s_1i_1 \times R_2s_1i_2 \times R_3s_1i_3 \times \cdots\cdots \times R_e \longrightarrow R$$
$$fc: R_1s_2i_1 \times R_2s_2i_2 \times R_3s_2i_3 \times \cdots\cdots \times R_e \longrightarrow R$$
$$fc: R_1s_3i_1 \times R_2s_3i_2 \times R_3s_3i_3 \times \cdots\cdots \times R_e \longrightarrow R$$
$$Given\ points\ of\ Intersection := function(fn)$$
$$fn_1 := Option \rightarrow p_1f_1j_1d_1t_1 |\ \text{Option} \rightarrow s_1i_1 |\ \cdots\cdots |\ Option(fn)$$
$$fn_2 := Option \rightarrow p_2f_2j_2d_2t_2 |\ \text{Option} \rightarrow s_2i_2 |\ \cdots\cdots |\ Option(fn)$$
$$fn_3 := Option \rightarrow p_3f_3j_3d_3t_3 |\ \text{Option} \rightarrow s_3i_3 |\ \cdots\cdots |\ Option(fn)$$

Figure 5. Pipeline Engineering Operations Translation Engine

Following the necessary structural framework that must be put in place for the language to implement its core operations, fragments of the syntactic elements representing the pipeline-built units are specified in BNF notation. The inclusion of this aspect is to ensure that the various pipeline build metrics are defined in the language [9]. The entire structure is a collection of pipeline components valuation functions represented as objects in the physical transmission mappings. The program therefore, is directly mapped into these pipeline attributes as the pre-processor. This is the point of harmonization between the translator in the translation scheme and the valuation functions of the semantic domain [24]. The semantic representations are such that for example, *component functionality* means that for every design operation: C = f(R), each pipeline component C will be oriented through the valuation function f to correspond to the value of the design operation. R represents the root (i.e., A Complete Pipeline) with the associated components, attributes and values. With a particular functionality, which describes a characteristic scenario of the pipeline build operation is such that $fc: C_1d_1t_1 \times C_2d_2t_2 \times C_3d_3t_3 \times \cdots\cdots \times C_e \rightarrow R$. Components $C_1$…………$C_3$ and their types $t_1$…….$t_3$ each with dimensions $d_1$ …… $d_3$, are functionally aligned together through the invocation of the valuation parameter from the concrete syntax. The process will continue to include the pipe component p, which is usually the major pipeline physical component involved in any oil ang gas pipeline engineering project. The pipeline network will now incorporate fittings, fitting types $ft_1$…...$ft_n$ and fiting dimensions $fd_1$…...$fd_n$. It will simialarly include loints, joint types $jt_1$…...$jt_n$ and joint dimensions $jd_1$…...$jd_n$. The pipeline with all the component types and dimensions will then be reinforced with supports $s_1$ …...$s_n$ and instrumentation $i_1$……$i_n$. The system can now allow for stakeholder variabilities inclusion with $fn_1 := Option \rightarrow p_1f_1j_1d_1t_1 |\ \text{Option} \rightarrow s_1i_1 |\ \cdots\cdots$

| $Option(fn)$ such that each design scenario from any stakeholder view point can be captured in the design sequence. This process continues un till the entire pipeline R for that particular pipeline project build operation is completed.

## 4. The Valuation Implications

This section presents examples that makes use of the valuation commands in the system [12]. The size and complexity are scripted and the code snipets illustrated as an industry piece that can be extended for the design of large complex pipeline systems. For example, as shown in the snipet: clc Ce=input ('Please select the design operation. Enter 1 for Pump selection, 2 to calculate the system total loss, and 0 to exit:'); %% At this point, we called the design operation, Ce. When the user enters the values 1, 2, 0 as necessary from the multiple document interfaces (MDI), a modeling action is initiated as seen in this code section: switch Ce case 1 %%This executes on pump selection request selection=input ('Let us start building the different Routes. Enter 1 to continue or press 2 to exit'); %% The different possible Routes are taken into account here switch selection case 1 %%The user is given the option of selecting a component or a %%joint p=input ('Please enter 1 for components, 2 for joints and 0 to complete/exit'); Once done, the call sequence corresponding to the selected value from the domain experts view point is validated and input parameters captured [25]. After the system script is run, a route selection prompt will enable the user to follow a guided sequence in order to display a design intent as shown:

```
Please enter the elbow length (mm):15
Please enter the elbow minor loss coefficient. Please note that the minor loss for elbows
Please enter the internal diameter of the Pipe (inches):24
Please enter the Pipe length (mm):12
Please enter the Tee length (mm):14
Please enter the Tee minor loss coefficient. Be aware of the Tee types:0.3
The Total length of the system is (mm): 53
Please enter desired velocity (mm/s):23
Please enter the fluid density (kg/mm^3)0.8
Please enter the fluid specific gravity0.008
Please enter the relative roughness of the pipe (mm):0.5
Please enter the static head of the fluid (mm):12
```

The system will then load from the data model into the view logic for the necessary modelling actions to be taken [12]. Depending on the model selection and view points broadcasting, a system curve will be displayed to showcase exact and intended designs as shown in figure 6.

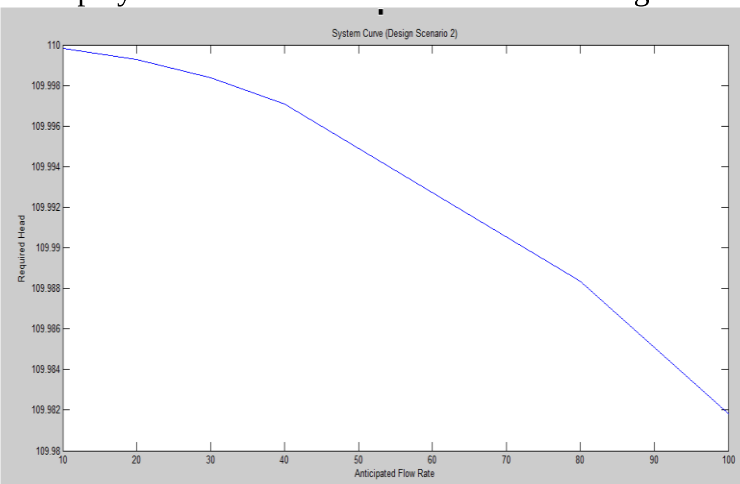


Figure 6 System Curve for a Particular Design Scenario

Quite a lot of decisions as clearly illustrated can be deduced from each activity depicting a typical pipeline construction activity where stakeholders need to have some prior knowledge before embarking on the project. The illustrations are as follows:

```
This pipeline system requires a Pump head of at least (in mm) 25.295
Please enter 1 to develop a system curve for the pipeline or 2 to exit1
Please enter all the anticipated flow rates over the system design life:[60 50 40 30 20]
Please enter 1 to develop a system curve for the pipeline or 2 to exit1
Please enter all the anticipated flow rates over the system design life:[100 90 80 70 60 50 40 30 20 10]
Please enter 1 to produce a Best Efficiency Point (BEP) or 2 to exit1
```

During these operations the pipeline engineer often referred to as the stakeholder can now run the system by obeying simple prompts to come up with more system curves that pertain to different design scenario for best efficiency point as shown in figure 7 through the editor interface.

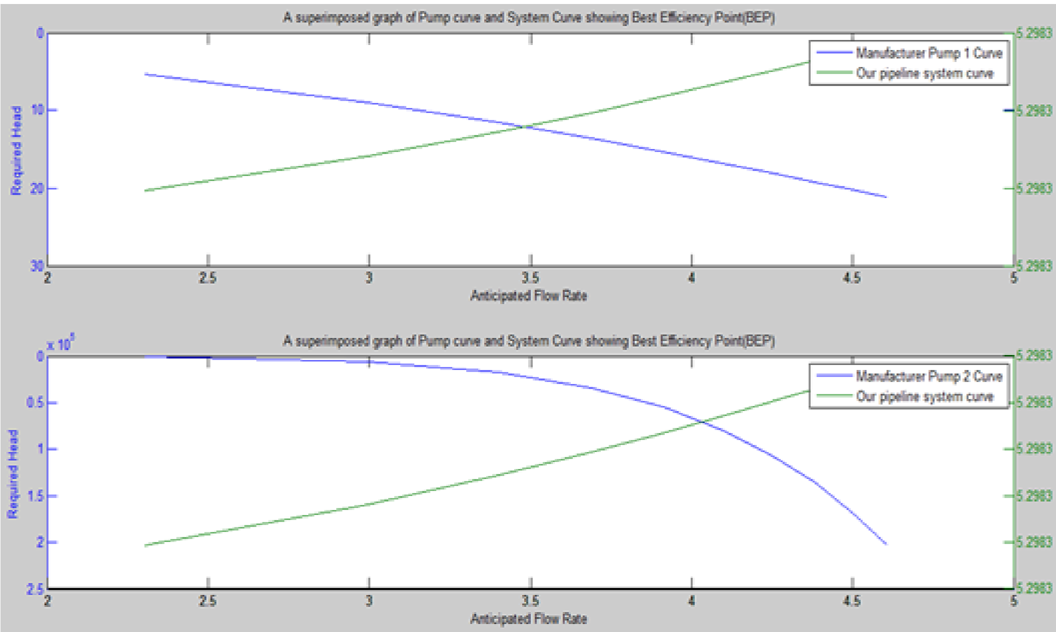


Figure 7 Design Scenario for Best Efficiency Point

Each design scenario adopted by the engineer is to solve a particular pipeline modelling intent that usually involve adequate and correct pipeline physical asset model selection, controls bothering on over sizing and under sizing of pumps, valves and other components and instrument, cost effectiveness, enhancement in operational productivity, and project quality control issues [26].

## 5. Conclusion

Issues of modeling engineering designs as practiced in the oil and gas industry, and particularly in oil and gas pipelines that convey gas or oil from sources to destinations are considered in this paper. It is practical that conventional computer aided design (CAD) software are indispensable tools in the pipeline engineering work environment. But most pipeline engineers find it worrisome to learn, understand and utilize conventional software in their line of business. This paper therefore, presented a model driven valuation technique that enable the engineer simply input values corresponding to familiar notations such as pipe diameter, fittings, flow metrics etc.) on an interface to get the kind of design, simulation artifact and other pipeline configurations without having to use any CAD or related system. These very vital aspects of the design analysis reflecting the domain characteristics were integrated as concepts to the policies and mechanics of the software production.

## 6. References

[1] Eric, E. (2003), Domain-Driven Design: Tackling Complexity in the Heart of Software Addison Wesley, USA.
[2] Gustavo, C. M., Sousa, F. M., Costa, G., Peter, J., and Clarke, A. A. (2012), Model-Driven Development of DSML Execution Engines, *Proceedings of ACM Conference*, eduMRT ’12, Innsbruck, Austria, 112 -118.
[3] James, P., and Roggenbach, M. (2014), Encapsulating formal methods within domain specific languages: A solution for verifying railway scheme plans, Mathematics in Computer Science 8 (1), 11-38.
[4] Markus, V., and Sebastian, B. (2013), DSL Engineering *Designing, Implementing and Using Domain Specific Languages* http://dslbook.org
[5] Sousa, F. M., Costa, P. J., and Clarke, A. A. (2012), Model-driven development of DSML execution engines, in: Proceeding of the 7th Workshop on Models@run.time (MRT’12), 10-15.
[6] Matthew T. (2021). Assistive Technologies Set to Transform Pipeline Projects. Pipeline and Gas Journal December 2021, Vol. 248, No. 12
[7] Brady, E., and Hammond, D. K. (2012), Resource-safe systems programming with embedded domain specific languages, Practical Aspects of Declarative Languages, Journal of Computer Science and Applications, vol 7(9) 242-257.
[8] Crimi, C., Guercio, A., Nota, G., Pacini, G., Tortora, G., and Tuccit, M. (2012), Relation Grammars and their Application to Multi-dimensional Languages. *Journal of Visual Languages and Computing*, 2(4), 333-346.
[9] Ho, M.; El-Borgi, S.; Patil, D.; Song, G. Inspection and monitoring systems subsea pipelines: A review paper. *Struct Health Monit* 2019.
[10] Wright, R.F.; Lu, P.; Devkota, J.; Lu, F.; Ziomek-Moroz, M.; Ohodnicki, P. Corrosion Sensors for Structural Health Monitoring of Oil and Natural Gas Infrastructure: A Review. *Sensors* 2019, *19*, 3964.
[11] Richard N. (2021). Southern California Oil Spill: Unfortunate Harbinger for Future Industry Distrust Pipeline and Gas Journal December 2021, Vol. 248, No. 12
[12] Vahdati, N., Wang, X., Shiryayev, O., Rostron, P., & Yap, F. F. (2020). External Corrosion Detection of Oil Pipelines Using Fiber Optics. *Sensors (Basel, Switzerland)*, *20*(3), 684. https://doi.org/10.3390/s20030684
[13] Popescu, C., & Gabor, M. R. (2021). Quantitative Analysis Regarding the Incidents to the Pipelines of Petroleum Products for an Efficient Use of the Specific Transportation Infrastructure. *Processes*, *9*(9), 1535. MDPI AG. Retrieved from http://dx.doi.org/10.3390/pr9091535
[14] Ramírez-Camacho, J.; Carbone, F.; Pastor, E.; Bubbico, R.; Casal, J. Assessing the consequences of pipeline accidents to support land-use planning. *Saf. Sci.* 2017, *97*, 34–42.
[15] Khan, F.; Yarveisy, R.; Abbassi, R. Risk-based pipeline integrity management: A road map for the resilient pipelines. *Pipeline Sci. Eng.* 2021, *1*, 74–87.
[16] Green, K.P.; Jackson, T. Pipelines Are the Safest Way to Transport Oil and Gas. Fraser Research Bulletin by the Faser Institute, Canada. 2015. Available online: https://www.fraserinstitute.org/article/pipelines-are-safest-way-transport-oil-and-gas (accessed on 3 May 2021).
[17] Anderson, D.A. Natural Gas Transmission Pipelines: Risks and Remedies for Host Communities. *Energies* 2020, *13*, 1873.

[18] Anvil International. (2012), Pipe Fitters Handbook, University Park, IL United States.
[19] Deng, J., Hormann, K., and Kazhdan, M. (2012), Geometric modeling and processing, Computer Aided Geometric Design 29 (7), 4-21.
[20] Cech, M.; Davis, P.; Guijt, W.; Haskamp, A.; Huidobro Barrio, I. Performance of European cross-country oil pipelines. In *Statistical Summary of Reported Spillages in 2019 and Since 1971*; Report CONCAWE: Brussels, Belgium, 2021; No. 4.
[21] Zhang, P.; Chen, X.; Fan, C. Research on a Safety Assessment Method for Leakage in a Heavy Oil Gathering Pipeline. *Energies* 2020, *13*, 1340.
[22] Adegboye, M.A.; Fung, W.-K.; Karnik, A. Recent Advances in Pipeline Monitoring and Oil Leakage Detection Technologies: Principles and Approaches. *Sensors* 2019, *19*, 2548.
[23] Martin Fowler (2020) Domain Driven Design DomainDrivenDesign (martinfowler.com) 22 April 2020
[24] Japheth, B. R., Osaisai, E. F. and Orukari, M. (2021) Controlling the Design Dynamics of Oil and Gas Pipelines for Effective Fluid Flow Using Domain Specific Modelling. International Journal of Basic Science and Technology ISSN 2488-8648 September, Volume 7, Number 2, Pages 169 – 176 http://www.ijbst.fuotuoke.edu.ng/ 169
[25] Japheth, B., R. Evans, F. O., Joseph A. E., and Juliana I. C. Mathematical Model of DSM Tool for The Design and Modelling of Fluid Transportation Pipeline. African Journal of Engineering and Environment Research (AJOEER) Volume2 Issue2, July2021 doi.org/10.37703/ajoeer.org.ng/q2-2021/01
[26] Bunakiye R. Japheth, Erho A. Joseph and Evans F. Osaisai, "A Survey of Model Selection for Oil and Gas Pipeline Design using Domain Specific Modelling'', African *Journal of Computing & ICT*. Volume 13, Number 3, September 2020.