Oil and Gas Pipeline Design Management System: A Case Study for Domain-Specific Modeling

Japheth Bunakiye Richard[1], Asagba Oghenekaro Prince[2]

[1]Department of Mathematics/Computer Science, Faculty of Science, Niger Delta University, Yenagoa, Nigeria
[2]Department of Computer Science, Faculty of Physical Science and Information Technology, University of Port Harcourt, Port Harcourt, Nigeria

Email address:
rb.japheth@ndu.edu.ng (J. B. Richard), asagba.prince@uniport.edu.ng (A. O. Prince)

Abstract: The intent of this paper is to define a common case study for the domain-specific modeling research community. The domain of the case study is oil and gas pipeline design management systems, i.e., systems 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, by allocating and managing resources, and by providing access to relevant design-related information to domain experts. This document contains general informal requirements for a pipeline design management systems (PDMSs), 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.

1. Introduction

The need for cost-effective and efficient design tools and methodologies that could aid better, and provide faster and productive solutions to pipeline designs has grown significantly over time. A pipeline design can range from interactive aggregation of graphics primitives to the formation of a graphics model through assemblies and subassemblies of the primitives in a typical CAD system [1]. Applying the principles of solid and parametric modeling, these graphics models can be built up to possible display in various ways and also to determine some material properties, they can even be assigned parametric dimensions and geometric constraints to define features, and to create relationships between these features in order to create what is referred to as intelligent models [12]. Though these intelligent models can be functional to determine the analysis of the application of the properties of the members of the oil and gas pipeline domain of engineering activities such as withstand external constraints, flow dynamics, applied loads, temperature and pressure, the process is still classified as a repetitive and time wasting task.

The reason for the repetition is that objects are explicitly described in conventional design modeling systems (i.e. Computer Aided (CAD) Systems). When one aspect of the model is changed, often several changes have to be made to satisfy design intent or the implicit rules of the design. This is because the software [1] does not keep track of the rules and the designer must decide where and when they are broken. The challenge is the issue of interaction between models, interactions in the way of concepts devoid of possible parametric constraints within a CAD system [10]. 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.

Domain specific modeling is predicated on seeing the pipeline design (i.e. the graphics model from a CAD) as the entity during development, it is the model that reflects the prescriptive technical characteristics prevalent in the domain. It also represents the concepts of the domain within which

language formalism is created to enhance model interaction and processing to produce an executable program or other models [10]. The aim is to abstract away technical details of computing from the domain engineer, allowing them to create pipeline designs or specifications without having to be an expert programmer or user of CAD systems. To make sure that the domain-specific modeling approach is somehow applicable to this case study, we present a collection of models from AutoCAD that describe the components of a pipeline design at different levels of abstraction [12].

2. Related Work

Domain-Specific Modeling raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. So generating some final products from high-level specification abstractions is a worthwhile software engineering venture for productivity. In this same line of thought Kienzle et al. [3], defined a common case study for the aspect-oriented modeling (AOM) research community. In their work, the domain of the case study is crisis management systems (CMS), i.e., systems that help in identifying and handling a crisis situation by orchestrating a communication and allocating managing resources between all parties involved in handling the crisis. Their goal was to making sure that all AOM approaches and techniques are somehow applicable to this case study, by presenting a collection of models that describe the CMS at different levels of abstraction. Afredo et al. [14] in their case study defined the requirements of a Software Product Line (SPL) aimed at managing car crash crises. Basic features along with desired variations were proposed such that it results in a small SPL definition. The primary focus of their proposed variations was to allow for static and dynamic variations (i.e., dynamic change between variants at runtime). All the information concerning possible variations and their possible implementations were introduced, which serves to illustrate the individual advantages and disadvantages of aspect-oriented modeling (AOM), feature-oriented models (FOM), and object-oriented modeling (OOM). This work adopted the same approach to defining a case study for the domain specific modeling (DSM) research community. The domain of our case study is pipeline systems design management and is closely tied to the DSM manifesto, which is, raise the abstractions; such that a language metamodel has to represent concepts from the domain of consideration, i.e. the domain of oil and gas pipeline engineering.

3. Methodology

We are basically looking at requirements in a pipeline systems modeling system. Our approach is hinged on user requirements documentation for a pipeline systems modeling system based on the theory and practice of domain specific modeling. In line with Domain specific modeling guidelines, we have to as a first step visit an industry for a case study. BG Technical Nigeria limited was visited for over a period of three months and the user requirements as real requirements analysis

document was created.. BG Technical Limited (BGT) [11] is a pipeline service company located in Nigeria with significant activity in Africa. The key requirements for the system work flow are as follows [9]:

1 Stakeholders design intent characterized as the view points of the input parameters should be part of the design concept. The term design intent refers to identified interests and disciplines of all the stakeholders of this system as it relates to pipeline design and highlights physical attributes that must be considered in completing the modelling process. Physical attributes are those parameters that govern the size, layout, and dimensional limits or proportions of the pipeline components. The components here refer to the graphics models produced from primitives commonly found in CAD systems.

2 To ensure that the graphics models represents the pipeline domain model with sound underlying pipeline engineering principles, which can be linked to produce a total life cycle approach to pipeline systems design and operation.

3 To be embedded in the domain model a semantic model subset with a focus on the user's perspectives, and consisting of the classes of the concepts and their relationships. As knowledge changes, the semantic model can change too, to ensure that physical components continue to do what the users want them to do in order to preserve and to produce clear design specifications for pipeline physical assets such as pipes, valves, active equipment (pumps, compressors, etc.,), instruments and supports.

4 Functional commonalities of the application domain should be identified in the modelling language structure. These common and differing functions can be abstracted and represented so that specific design scenarios are evolved as an adaptation or refinement of the model.

5 A domain analysis module should be incorporated with a feature model representing the physical components to support communication between the requirements analysis and the design phases respectively.

4. Contribution

Pipeline Design is the process of creating detailed plans and drawing of the nature of the pipeline with a view to solving problems that might occur in the construction and operation of the pipeline these problems may be hydraulic, structural or geotechnical. Pipelines are the most common means of transporting oil or gas [11]. A pipeline is like any other flow line. The main differences are that pipelines are long and continuously welded; they are most often either buried or otherwise inaccessible due to their location over the majority of their length. These differences mean that small sections of pipeline are not easily removed for maintenance and consequently great care is taken to prevent problems arising in the first place [10]. Also pipelines are extremely expensive to lay, and as such great care must be taken to put the design from

inception in the right perspectives and on time. Generally, when designing a pipeline, the engineer considers the physical and chemical properties of the fluid, to be pumped through the pipeline; the maximum volume of fluid that will be pumped through the pipeline at any time, the nature of the environment through which the pipeline is going to traverse; whether the pipeline is on land or offshore and the whether the climate is warm or cold and the required delivery pressure [4]. More specifically the engineer considers Pipe diameter required (The larger the diameter of the pipeline, the more fluid can be moved through it), Pipe length (The greater the length of a segment of pipeline, the greater the total pressure drop), Specific gravity and density Compressibility, Operating and ambient temperatures, Viscosity and Vapour Pressure etc. In the design of oil and gas pipelines, pressure drop, flow capacity and pumping or compression horsepower required are key calculations [4]. In most pipeline calculations, assumptions must be made initially. For instance, a line size may be assumed in order to determine maximum operating pressure and the pressure drop in a given length of pipe for a given flow volume. If the resulting pressure drop, when added to the known delivery pressure exceeds the allowable working pressure, a larger pipe size must usually be chosen [9].

4.1. Materials for Adaptability

The pipeline design management system should contain the following functionalities for adaptability with conventional DSM platform:

a. A rule processing module responsible for coordinating the communication between graphics models specific to transmission pipelines and a layer of reusable software interface crafted to handle modeling in a timely manner [11].
b. The user could make some input through guided notations from an interface, and the system can then match these inputs with a library of domain concepts to produce desired designs.
c. Creating models that can be processed to produce artefacts.
d. Wrap up and archive designs to produce target codes

4.2. Materials for Usability

The pipeline design management system shall exhibit the following non-functional properties for usability:

a. Screen Organization
 - The sequence of the screen items should not be confusing.
 - The system shall not contain characters in the screen that are hard to read.
 - The system shall provide support for highlighting that simplifies text.
 - Organization of information should not be confusing.
b. Terminology and System Information
 - Use of terms throughout the system should be consistent.
 - Terminology used in the system should always relate

to task.
 - Prompts for input should be clear.
 - The system shall be able to inform about its progress.
 - The system shall provide error messages that are helpful.
 - Position of messages on screen shall be consistent.
c. Learning
 - Learning to operate the system shall be easy.
 - The system shall provide support for remembering names and use of commands.
 - Performing tasks shall always be straightforward.
 - Supplemental reference materials shall be clear.
 - Help messages on the screen shall be helpful.
 - Exploring new features by trial and error shall be user friendly.
d. System Capabilities
 - System speed shall be fast
 - The system shall be reliable enough to provide interaction between modules.
 - The system shall be stable.
 - The system shall provide support for easy correction of mistakes.
e. Adaptability
 - The system shall provide alternate strategies for dealing with design intent.
 - The system shall showcase all familiar notations.
 - The system shall be able to maintain effective design artefacts.
f. Accessibility
 - The system shall support all input requesting for resource at a time.
 - The system shall support coordination, and information access.
 - The system shall support management of all design intent at a time.
 - The system shall support management of designs at all times.
g. Real time
 - The control of artefact orientation shall be updated.
 - The system shall be able to retrieve any stored information promptly.

4.3. Representing Domain Knowledge

The models presented in this paper focus particularly on oil and gas pipeline designs. Based on domain-specific modeling; the PDMS includes all the functionalities of general domain-specific modeling languages, to this end, the models must represent things in the pipeline engineering domain so that the determining factors in the pipeline design management system can be accommodating in order to come up with an optimal solution [11]. Critical in the determining factors are the stakeholders; stakeholders are the domain experts and related users of the system whose design intent is characterised as the viewpoints of the input parameters [1]. There are competing design requirements among stakeholders; each one has their own set of constraints, objectives and responsibilities. Whereas stakeholders' objectives describe the bit of problem(s)

addressed by the typical modelling tool, the responsibilities describe associated design intents [4]. The stakeholders who are the actors are defined along their lines of interest on design bases such as the physical attributes, loading and service conditions, and environmental and materials-related factors [6]. Physical attributes are those parameters that govern the size, layout, and dimensional limits or proportions of the pipeline.

1. Stakeholders in the aspects of Loading and Service Conditions- Stakeholders in this category are typically interested in working on pipeline designs with parameters such as forces, pressure changes, temperature changes, thermal gradients, or any other parameters that affect the state of stress of the piping system [3]. Their main objectives are:

1   to ensure that loading conditions are both internally and externally clearly specified,
2   to clearly describe service conditions as combinations of loads,
3   to state accurate estimation of dimensions of resources needed,

To achieve these objectives, their responsibilities are:
1   to determine what, and how many parameters to be encoded,
2   to propose a strategy for handling the pipeline design process,
3   to specify appropriate pipeline design codes and standards,
4   to provide the designer with guidance in setting appropriate design stress limits.
5   to state clear, executable instructions to appropriate staff.

2. Environmental Factors- Stakeholders concerned with environmental factors in a pipeline are interested in pipeline design products with features which can ultimately lead to a breach of the pressure boundary or a gross structural failure [3]. Their main objectives are:

1   to determine appropriate pipe bed during design,
2   to define adequacy of performance of the pipeline system,
3   to list sufficient environmental hazards.

To achieve these objectives, their responsibilities are:
1   to give concise localized information on the environment,
2   to assist in materials selection,
3   to ensure maintenance of the pressure integrity of a piping system, within predefined criteria limits.

3. Use of Codes and Standards in Pipeline Design- Stakeholders in this category are most widely interested in making sure that a pipeline design adheres to required standards. Their main objectives are:

1   to verify that piping codes provide specific design criteria such as permissible materials of construction, allowable working stresses, and load sets that must be considered in design,
2   to certify that piping codes that relates to pipeline design such as American Society of Mechanical Engineers [4 (ASME B31.1, ASME B31.3)] are followed.

To achieve these objectives, their responsibilities are:
1   to give accurate and detailed records on standards and codes,
2   to source for recent sources of publications on piping

standards and codes,
3   to give designers clear view on the difference between standards and codes.

4. Piping Joints- Stakeholders in this category are interested in major impact on the initial installed cost, the long-range operating and maintenance cost, and the overall performance of the piping system. Their main objectives are:

1   to state the physical design attributes of the joints,
2   to specify the relationship criteria between joints,
3   to identify the components in a joint,
4   to define the type of units (e.g. metric).

To achieve these objectives, their responsibilities are:
1   to determine the joint location,
2   to verify the pipe section,
3   to determine the type of fittings,
4   to put in place factors necessary for joint selection and design.

5. Relative Anchor Movements- Stakeholders in this category usually valued the inclusion of support for a pipeline system to function properly.

Their main objectives are:
1   to state the physical design attributes of the supports,
2   to specify the relationship criteria between joints and supports,
3   to identify the components in a support,
4   to define the type of units for pipe size (e.g. metric).
5   to define the type of units for fluid flow design and pressure-integrity design (e.g. metric).

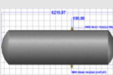To achieve these objectives, their responsibilities are:
1   to determine the support positions,
2   to verify the pipe size and support sections for fittings,
3   to determine the type of supports,
4   to put in place factors necessary for support selection and design.

5. Model Selection for Pipeline Physical Models

The DSM approach requires models representing things about a domain. Our domain here is the pipeline engineering that concerns with transmission of fluids. To this end, we are adopting CAD models to represent pipeline physical components in the domain. The pipeline physical components are referred to as the design models for the pipeline design management system and are given in this section [1]. Such components include pipes, flanges, fittings/joints, bolting, gaskets, valves, and the pressure components. Also included are pipe support components. The pipeline domain concepts are clearly structured from the attributes of these components, which invariably forms the library framework and vocabulary that maps the concepts to appropriate abstraction levels within the modeling system metamodel. In this context, the vocabulary necessary for the subsequent system specification can be captured from the components as a suit of AutoCAD objects [12]. The design models, therefore, are AutoCAD objects that depict the typical pipeline fundamentals and

materials, which form the instance of the pipeline design management system. A cross section of some of the design models are shown in table 1.

Table 1. Typical Pipeline Design Models.

Component	Design Model
Pipe Cross Section	
Bolt and Nut	
Flanges	
Tank	
Supports	
Valve	
Reducer Fitting	
Elbow Fitting	
Tee Fitting	
Grooved Joint	
Butt Joint	
Meters/Gauges	

5.1. Feature Extractions

Any domain specific modeling system has a common set of responsibilities and functionalities, and can be applicable to a broad range of domains [2]. However in the pipeline engineering domain the common set of responsibilities and functionalities are based primarily on the physical components that form the building blocks of a pipeline. It is, therefore, natural [8] to build a framework or software product line of pipeline design management systems, which

can be specialized to create systems for a particular kind of design viewpoint and a particular context. A feature diagram (feature model) listing many possible features of a pipeline design management system is given in figure 1 [3]. The features represent the characteristically distinctive aspect of the different pipeline physical components for definition, organization and display of design data. The feature model is a tree structure, with features forming nodes of the tree. The root node represents the complete pipeline build concept. The features are hierarchically arranged with relationships between a parent feature and its child features categorized into AND, mandatory and optional nodes with the arcs and groupings of features representing feature variability and commonalities [2]. The variability indicates precisely what evidence is needed to specify an instance of a feature that could fit into a particular pipeline design event [5]. The commonalities in this respect define the set of common operations and primitives of the system.
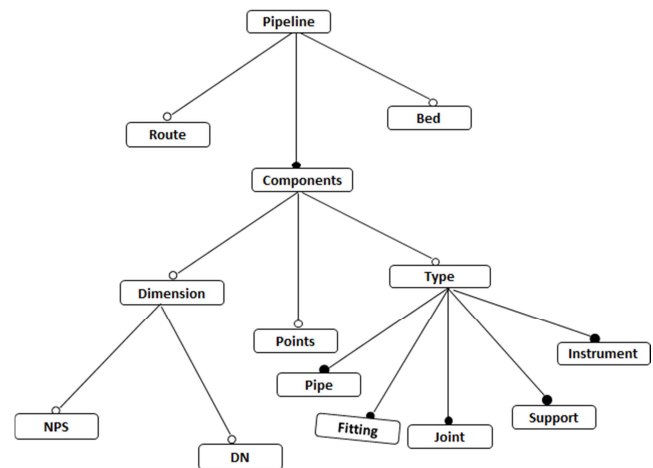

Figure 1. Feature Model Parent - Child Relationship.

Selection of some features requires the selection of other features. Examples of such dependencies are characterised in a feature relationship grammar as follows:
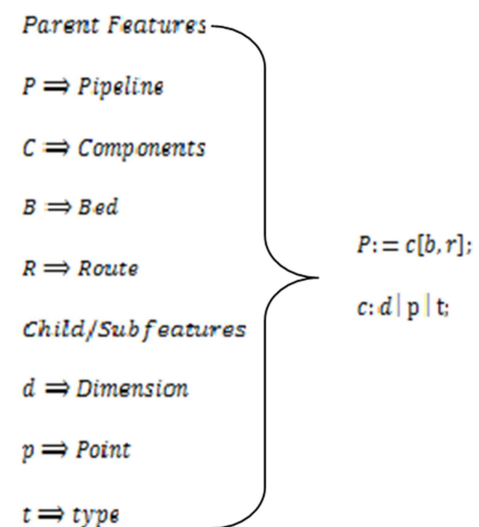

Parent Features

$P \Rightarrow Pipeline$

$C \Rightarrow Components$

$B \Rightarrow Bed$

$R \Rightarrow Route$

Child/Subfeatures

$d \Rightarrow Dimension$

$p \Rightarrow Point$

$t \Rightarrow type$

$P := c[b, r];$

$c : d \mid p \mid t;$

Figure 2. Parent - Child Relationship Grammar.

The feature diagram defines the pipeline design standard reference attributes and relationships of the product family. The product family defines the pipeline components p, f, j, i, s, and an associated tree grammar as shown in figure 2 above. From the grammar b is an optional feature; so optional child features of b, may or may not be included in the configuration of the design system. The other parent feature such as component is a compound mandatory feature that consists of optional features d, p and t, which means that D, P, and T must be selected optionally to be included in the configuration if and only if p, f, j, i, and s is included in the configuration [8].

5.2. The Metamodel Aspect

The domain model offers insight into the problem domain, in our case, the pipeline design management system [7]. Taking the form of expressions as refinements; sub setting the semantic space of the C# base language, the internal part of it is built on the DSL processor engine that compiles the DSL Builder files at the core of Microsoft DSL tool [13]. The domain model illustrated in figure 3 provides a description of the concepts of the problem domain relevant to the PDMS.
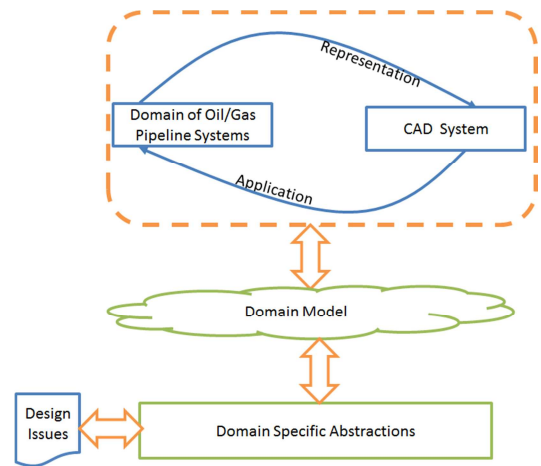

Figure 3. The Domain Model.

Although any domain concept could be added to the domain model, we decided to include here only concepts that must define information that must be recorded for the purpose of fulfilling the system's responsibilities over time. In other words, the domain model presented here only contains concepts that are used to describe the necessary information to fulfil system goals [6]. Figure 4 is the metamodel aspect representing the concepts as classes, attributes, generalization/specialization hierarchies and associations inherent in the domain of the PDMS [13].
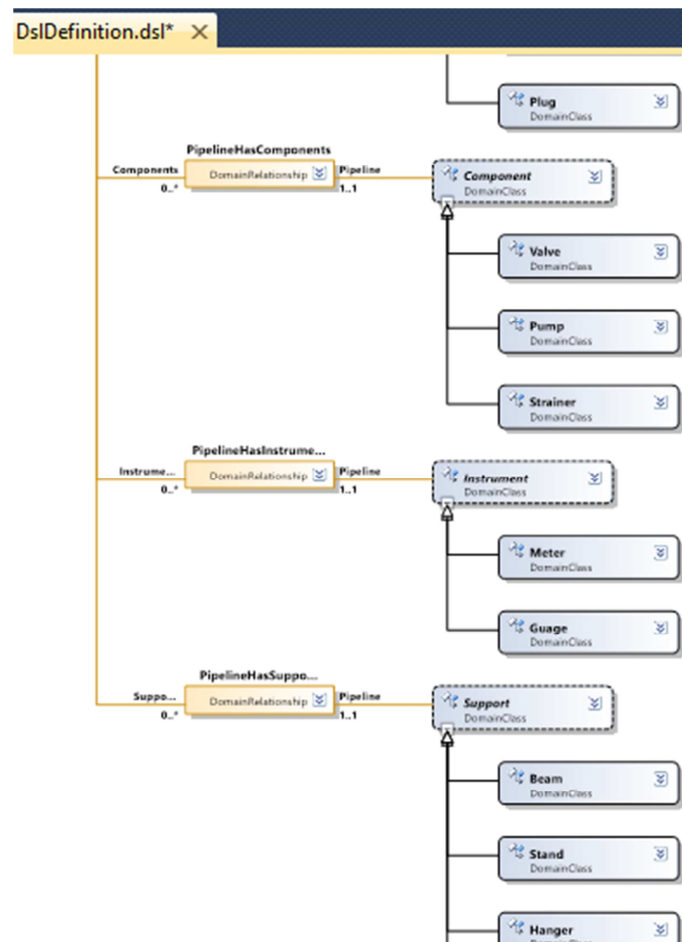

Figure 4. Metamodel.

6. Informal Physical Designs

Physical components in a pipeline project includes pipe, flanges, fittings and joints, bolting, gaskets, valves, and the pressure containing portions of other piping components [8]. It also includes pipe hangers and supports and other items necessary to prevent over pressurization and overstressing of the pressure-containing components [12]. It is evident that pipe is one element or a part of piping. Therefore, pipe sections when joined with fittings, valves, and other mechanical equipment and properly supported by hangers and supports results into a pipeline system as shown in figure 5.
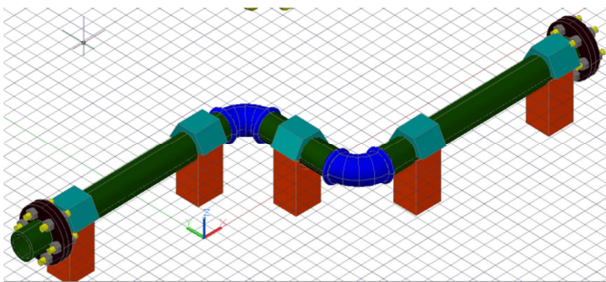

Figure 5. *A Typical Pipeline System.*

7. Conclusion

The domain specific modeling (DSM) approaches and techniques are meant to be used during different phases of software development. As a result, the DSM approaches work with different kinds of models and modeling notations. To make sure that the DSM approaches and techniques are applicable to a case study, we present a collection of models that describe the management of the design of a typical oil and gas pipeline system at different levels of abstraction. The models include a feature model representing the software product line, an informal physical pipeline model, and a domain model. The domain model, which offers insight into the problem domain, takes the form of expressions as refinements describing the concepts of the problem domain relevant to the pipeline design management system.

References

[1]   Autodesk Inc. (2013) AutoCAD Release 2013 Programmers Reference Manual.

[2]   Feature-Oriented Domain Analysis (FODA http://www.sei.cmu.edu/reports/90tr021.pdf.

[3]   Kienzle J¨org, Nicolas Guelfi, and Sadaf Mustafiz Crisis Management Systems: A Case Study for Aspect-Oriented Modeling Transactions on AOSD VII, LNCS 6210, pp. 1–22, 2010. Springer-Verlag Berlin Heidelberg.

[4]   Nayyar, Mohinder L. Piping handbook / [edited by] Mohinder L. Nayyar.—7th ed. p. cm. ISBN 0-07-047106-1 McGraw-Hill 2000.

[5]   Zekai Demirezen, Marjan Mernik, Verification of DSMLs Using Graph Transformation: A Case Study with Alloy, MoDeVVa'09, Denver, CO, United States, 2009 ACM 978-1-60558-876-6/09/10.

[6]   Assessing Composition in Modeling Approaches Gunter Mussbacher1, Omar Alam2, CMA'12, September 30 2012, Innsbruck, Austria Copyright © 2012 ACM 978-1-4503-1843-3/12/09 …$15.00.

[7]   D. Méndez Fernández, D. C. Petriu, N. Rouquette, Ø. Haugen (Eds.): A Meta Model for Artefact-Orientation: Fundamentals and Lessons Learned in Requirements Engineering; MODELS 2010, Part II, LNCS 6395, pp. 183–197, 2010 Springer-Verlag Berlin Heidelberg 2010.

[8]   Markus Voelter, Eelco Visser: Product Line Engineering using Domain-Specific Languages Independent, acm.org 2011 15th International Software Product Line Conference 978-0-7695-4487-8/11 2011 IEEE DOI 10.1109/SPLC.2011.25.

[9]   Bashar Nuseibeh, Steve Easterbrook: Requirements Engineering: A Roadmap... Department of Computer Science, Imperial College.

[10] Engineering & Piping Design Guide Manual No. E5000, October 1, 2007 National Oil well Varco Company, San Antonio, Texas, http://www.starfiberglass.com.

[11] B. G. Technical LTD - B. G. Technical Oil & Gas industry Port Harcourt, Nigeria. www.bgtechnical.com/Annual Reports 2009 to 2012.

[12] Alessandro NADDEO, Cad Active Models: An Innovative Method in Assembly Environment, Journal of Industrial Design and Engineering Graphics Volume 5 Issue No. 1 – 2010.

[13] Steve Cook, Gareth Jones, and Stuart Kent, (2007) Domain-Specific Development with Visual Studio DSL Tools, Pearson Education, Inc, USA.

[14] Afredo Capozucca, Betty H. C, Cheng, Geri Georg (2013) Requirements definition document for a software product line of car crash management system.