- ORIGINAL ARTICLE -

A Model of Reusable Assets in AIE Software Systems

Un Modelo de Activos Reusables dentro de la Ingeniería de Sistemas de Inteligencia Artificial

Alejandra Cechich[1], Agustina Buccella[1], Carolina Villegas[1], Ayelén Montenegro[2], Angel Muñoz[2], and Andrea Rodríguez[2]

[1]*GIISCO Research Group, Departamento de Ingeniería de Sistemas, Facultad de Informática, Universidad Nacional del Comahue, Neuquen, Argentina*
{alejandra.cechich,agustina.buccella}@fi.uncoma.edu.ar
[2]*Instituto Nacional de Tecnología Agropecuaria (INTA), Alto Valle de Río Negro y Neuquén,*
{*montenegro.ayelen,munoz.angel,rodriguez.andrea*}@*inta.gob.ar*

Abstract

Nowadays, due to the increasing presence of artificial intelligence in software systems, development teams face the challenge of working together to integrate tasks, resources, and roles in a new field, named AI Engineering. Proposals, in the way of models, highlight the needs of integrating two different perspectives – the software and the decision-making support (big data, machine learning, and so on) systems. But there is something more – both systems must achieve high quality levels for different properties; and this is not a straightforward task. Quality properties, such as reusability, traditionally evaluated and reinforced through modeling in software systems, do not exactly apply similarly in decision-making support systems. In this paper, we propose a model for managing reusable assets in AI engineered systems by linking software product line modeling and variety identification. The proposal is exemplified through a case study in the agriculture domain.

Keywords: Big data variety, Software product lines, Data Analytic, Reusability

Resumen

Hoy en día, debido a la creciente presencia de la inteligencia artificial en los sistemas de software, los equipos de desarrollo enfrentan el desafío de trabajar juntos para integrar tareas, recursos y roles en un nuevo campo, denominado Ingeniería de IA. Las propuestas, en forma de modelos, destacan la necesidad de integrar dos perspectivas diferentes: el software y los sistemas de apoyo a la toma de decisiones (big data, aprendizaje automático, etc.). Pero hay algo más: ambos sistemas deben alcanzar altos niveles de calidad para diferentes propiedades; y esto no es una tarea sencilla. Las propiedades de calidad, como la reutilización, tradicionalmente evaluadas y reforzadas a través del modelado en los sistemas de software, no se aplican exactamente de manera similar en los sistemas de apoyo a la toma de decisiones. En este artículo, proponemos un modelo para administrar activos reutilizables en sistemas de ingeniería de IA al vincular el modelado de líneas de productos de software y la identificación de variedades en sistemas de análisis de datos. La propuesta se ejemplifica a través de un estudio de caso en el dominio agrícola.

Palabras claves: Variedad en Big Data, Líneas de Productos de Software, Analítica de Datos, Reusabilidad

1 Introduction

Traditional software and decision-making support (DMS) systems must collaborate each other to reach the goal of having efficient AI engineered (AIE) software systems, which should be able to process a huge amount of data as well as performing traditional organizational operating processes [1]. Developing both perspectives is like thinking of a two-side conceptualization. On one hand, the software system must satisfy operational stakeholders' needs; and, on the other hand, the DMS system (by using data analytic) must reach accuracy and precision for decision-making. Quality properties, such as reusability, are also addressed by the two systems; however, the different conceptualizations make researchers to rethink the way these properties should be modeled.

From a domain-oriented perspective, reusability has been traditionally modeled by using software product lines (SPLs) [2, 3, 4], where the domain acts as the center for designing components. Here, flexibility and reuse are supported through variability management, which includes mechanisms for defining, designing, implementing and testing variabilities. Then, products are generated as a result of all the commonalities and variabilities chosen by specific requirements of the product being developed. In previous work, we have proposed a domain-oriented development based on the use of domain taxonomies that guide the definition and reuse of services in SPLs. Our proposal focuses on characterizing geographic information systems because of several reasons. First of all, and as the geographic area has important advances on standardization, we took advantage of standards defined by the ISO/TC 211[1] (Geographic information/Geomatics) committee and the Open Geospatial Consortium[2] (OGC), and specialized them into specific subdomains, including marine ecology and paleontology [5, 6, 7]. Then, we produced a set of services able to be reused among different products, aiming at satisfying different (but in some way similar) software requirements.

By mirroring variability management and domain-oriented development in SPLs, we might conceptualize data variety for DMS systems (DMSs), including Big Data Systems (BDSs)[3], as a set of cases that should be abstracted according to their different varieties. In this way, reusable AIE software systems might be modeled as a two-side reusable view.

To the best of our knowledge, there are no proposals in the literature addressing reuse in the same way as we are proposing. For data analysis, reusabililty is analyzed through different points of view or focused on different parts of a system, for example, in [8, 9] authors analyze reusability centered on the use/reuse of data and/or open data. Other proposals are focused on reusability in terms of increasing collaboration when developing BDSs [10].Thus, our main contributions here are (1) a model for supporting the creation/use of reusable artifacts for DMSs development in a AIE software system, and (2) the instantiation of the model in the agriculture domain.

This article is organized as follows. In the next two sections we summarize related work and previous concepts/definitions. Then, we introduce our model for reusable AIE software systems, further discussing variety management in the DMSs side. Afterwards, a case study instantiates the model as a proof of concept. Finally, conclusions and future work are addressed in the last section.

2 Related work

In DMSs, including BDSs, reusability has been approached from various angles. For example, [9] discusses concepts of reusability in the context of data analytics, distinguishing between use and reuse of data. Various open research questions on reuse are proposed here, such as trade offs between collecting new data and reusing existing ones; the needs to distinguish between use and reuse; etc. More specifically, but in the same sense, [8] goes deeper into privacy aspects in the context of data reusability. There, a data reuse taxonomy is proposed, which may be useful to determine to what extent this reuse should be allowed, and under what conditions, to preserve privacy.

Other proposals address reuse aspects in terms of increasing collaboration in the development of BDSs through the use of new technologies (i.e. cloud computing). For example, in [10] an approach to managing BDSs through storage and processing capabilities in a public cloud is proposed and exemplified. Additionally, the different support platforms for the development of BDSs are addressed from a reuse point of view. For instance, the proposal in [11] analyzes improvements in the efficiency of tools, such as Apache Hadoop[4] and Spark[5], due to the reuse of artifacts between different projects. For this, common aspects are analyzed; and a workflow implemented in a scalable and extensible way is provided.

On the reuse management side, in [12] an exploratory analysis is carried out through interviews to *Microsoft* scientists, resulting in a set of reused tasks within a DMSs life cycle, together with the strategies to share and reuse previous work done in the same activity in a different project.

In addition to managing reuse across different teams and working with functions and code made by others, reusability was also addressed in terms of how architectures can be composed. In this sense, [13] incorporates the detection of common and variable aspects within the development of BDSs in order to build *families of systems*.This work introduces a reference architecture that allows system designers to: (1) *define requirements*, the reference architecture identifies significant requirements and shows variations according to the type of requirement; (2) *develop and test solutions*, the architecture identifies modules that must be developed in order to enable a certain required capability; and (3) *integrate systems*, existing systems can be mapped to the modules of the reference architecture, which leads to easy identification of points of conflict where interoperability between systems must be worked on.

This reference architecture is organized as a collection of modules that decompose the solution into elements that perform functions or capabilities for a set of *aspects*, such as “reusable modules”, representing the ability of the system to reuse. Examples include differences found in data entry rates, constrained by types of input streams or user requests; decisions on whether data ingestion is done in fixed periods or dynamically in real time; and so on.

The architecture is also mapped to use cases (strategic geospatial information analysis and visualization, signal analysis, etc.). Similar to our proposal, relevant requirements are identified, including categories such as data types (unstructured text, geospatial, audio, etc.), data transformation (clustering, correlation), data visualization (images, networks), etc. However, differently from this proposal, we define a two-side view of the

[1] `https://committee.iso.org/home/tc211`

[2] `http://www.opengeospatial.org/`

[3] In this paper, specific cases of DMS systems, such as big data systems, will be conceptualized similarly.

[4] https://hadoop.apache.org/

[5] https://spark.apache.org/

family of systems by explicitly considering both parts of an AIE software system. We built upon previous work on variability modeling in software product lines, and borrow documentation techniques to explicitly representing variety for DMSs. By doing so, we expect the similar documentation will contribute to easy understanding and smooth transition between both views.

Previous concepts and definitions are introduced in the next section to facilitate reading.

3 Fundamentals of a reusable model for AIE software systems

3.1 Software Product Line (SPL) System

For the Software Subsystem, variability management in software product lines has been extensively researched in the last years [14]. Variability is currently documented in several ways; therefore, the reusable software subsystem (SPL) may adopt any of these proposals. Particularly, we follow our own methodology that uses datasheets to document a functional and standard-based view of variability [5, 6].

Our approach emerges as a functionality-oriented design. That is, each functionality of the SPL is documented by a functional datasheet representing the set of services, commons[6] and variants, which interact to reach the desired functionality. Each datasheet is documented by using a template composed of six items containing an *identification*, such as a number or code; a *textual name*, describing the main function; *the domain* in which this functionality is included; the *list of services* required for fulfilling this functionality; a *graphical notation*, which is a set of graphical artifacts showing the service interactions (as common and variant services); and a *JSON*[7] *file* specifying the services and their interactions. Within the *graphical notation item* (of the datasheets) any graphical artifact might be used; and particularly, we use an artifact based on variability annotation of collaboration diagrams (of UML). The required variability, according to the functionality to be represented, is attached to the diagrams by using the Orthogonal Variability Model (OVM) notation. We named this artifact *variability model*.

Examples of datasheets can be found in [5, 6] for the marine ecology domain, and in [7] for the paleontology one. For instance, Table 1 shows the datasheet *Add New Excavations*, a variability example extracted from [7] to model processing variations of the *Load Excavation Data* service, which are identified by three possibilities (GPS, file, form).

3.2 Data variety identification

Data are diverse in terms of their structure, dimensions, way of collecting, etc. This property, usually referred as *variety*, is classified into for cases in [15]: structural, sources, content, and process. Data variety has been also considered from the point of view of incorporating semantics into the modeling process, even relating several quality properties, such as interoperability, security, reusability, etc. In previous work we have extended the classification of variety given by [15] to include context variety as part of the variety modeling process [16]. Let us briefly summarize the different varieties we can find as follows.

- Source variety refers to three types: data that are automatically generated by sensors, security cameras, intelligent agents that collect data from a vehicle, etc.; data from interactions between humans and computers, such as emails, document submissions, photo uploads, etc.; and data collected by business processes, which are generated by transactions and can include metadata regarding which data bases are accessed, etc. Data from these sources can be classified according to their different types and formats as (1) structured, (2) semi-structured and (3) unstructured. In class (1) are data that are extracted from relational databases, with well-defined formats; class (2) includes data that do not follow a restrictive tabular structure, but use labels to identify attributes, such as documents in XML or JSON; and class (3) includes data with no structure, such as plain text files, log files, data collected from social networks, etc.

- Content variety refers to data that can have a simple basic dimension, such as text, images, audio; data with multiple dimensions obtained by combining simple dimensions; or data with a graph structure. In addition, content refers to variables that are relevant to a particular domain or requirement, possibly related to a contextualization; for example, water quality parameters when predicting water turbidity.

- Process variety refers to the way data are manipulated, classifying into (1) batch, where large volumes of data are processed offline and generally in parallel; (2) interactive; (3) real time, as the data stream enters the system; and (4) processing on graphs that represent individual objects (nodes) and their relationships (arcs) to other objects. Process variety also refers to the type of data analysis requirement at hand, such as prediction, diagnosis, etc.; and the type of processing algorithms used for attending those requirements.

- Finally, context variety [16] refers to identify domain variations that may constrain or affect the results of the analysis. The different types of contexts depend on each domain; for instance, for hydrology systems, water bodies can be classified as rivers, lakes, seas, etc.; where water flows can

[6] Common services are services that will be part of every product derived from the SPL

[7] https://www.json.org/json-es.html

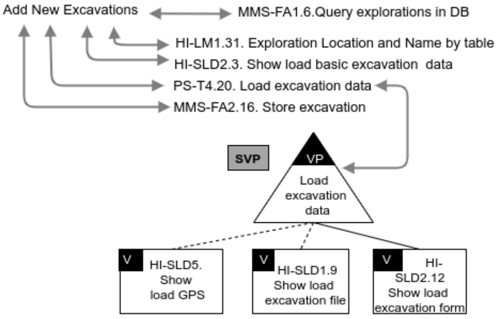
Id	FD2
Name	Add New Excavations
Domain	Palentology subdomain
Services	MMS-FA1.6, HI-LM1.31, PS-T4.20, HI-SLD2.3, ...
Variability Models	 Add New Excavations MMS-FA1.6.Query explorations in DB HI-LM1.31. Exploration Location and Name by table HI-SLD2.3. Show load basic excavation data PS-T4.20. Load excavation data MMS-FA2.16. Store excavation SVP VP Load excavation data V HI-SLD5. Show load GPS V HI-SLD1.9 Show load excavation file V HI-SLD2.12 Show load excavation form

Table 1: An example of the functional datasheet *Add New Excavations* (extracted from [7])

be influenced by climatology, geology, and so on. Therefore, structuring domain knowledge is a core issue when detecting context variety.

Variety may be identified during a data processing pipeline, from data ingestion to data analysis and visualization. The variety identification process (VIP) is split into two possibilities according to the working perspective of our proposal - top down and/or bottom up [16]. In a *top-down* perspective (T-VIP), given a domain problem, an expert user elaborates one or more hypotheses that should be tested through data analysis (i.e. Are data supporting this?); then, the hypotheses are taken by data analysts who proceed to work into the data processing pipeline; finally, results are returned to verify the hypotheses, possibly visualizing data in different ways, and allowing hypothesis reformulation or process ending, alternatively. On the other hand, a *bottom-up* perspective (B-VIP), given a domain problem, an expert user decides to launch an exploratory study to find out what data can reveal for this problem (i.e. What do data say?); then, the study is carried out by data analysts, again by applying the data processing pipeline; and finally, results (findings) are returned to be validated with experts, alternately ending the process or reformulating the search.

In our proposal, during the data processing pipeline, variety is influencing decisions and guiding these processes, so each activity and/or result might be reused. For instance, *source variety* during *data ingestion* will help detect different data structures, acquisition techniques, etc.; meanwhile, *process variety* will help detect variations in data analysis techniques; and finally, *context variety* will allow identify domain variations that may constrain or affect the results of the analysis during the whole pipeline.

We have proposed a simplified version of the datasheets, previously introduced, to document variety as knowledge assets. To do so, we have redefined the following concepts:

- *Variety Subject*: is a real-world variable element, including entities or actions; or a variable property of such an element.
- *Variety Object*: is a particular instance of a variety subject.
- *Variation Point*: is a representation of a variety subject along with domain artifacts enriched by context information.
- *Variant*: A representation of a variety object.

Variety within a domain can be defined in terms of the above concepts, generating artifacts named as *variety models*. As an example, we can think of the task *open the door* (VS: Variety Subject). This task has sub-cases (possibilities) such as *open the door with a key*, or *open the door with a reader card*. Here, the variation point (VP) is the *opening*, which can be of two types – with a key, or with a card reader (V: Variants). When the variation is mapped to a particular case (i.e. the office "A"'s door that opens by a lock with a key), the first variant is selected for this instance (VO: Variety Object).

As a data analysis variation example, in [16] we can see the case for *neural network*, with two different models available from the knowledge base implemented as CoVaMaT (Context-Based Variety Management Tool) [17]. Looking at the datasheet, we can see that there are two possibilities for these variety models: (1) optional variation points, when the proposal and existing models show similarities; and (2) alternative variation points, when a new model should be created and stored[8], or it is possible to reuse an existing one.

4 A proposal for reusable AIE software systems

Coming back to building a AIE software system, its analytics (DMSs) usually interrelates to services provided by traditional software systems, and particularly by reusable ones, such as software product lines. The

[8] The <<require>> restriction implies that every new model is stored in the KB

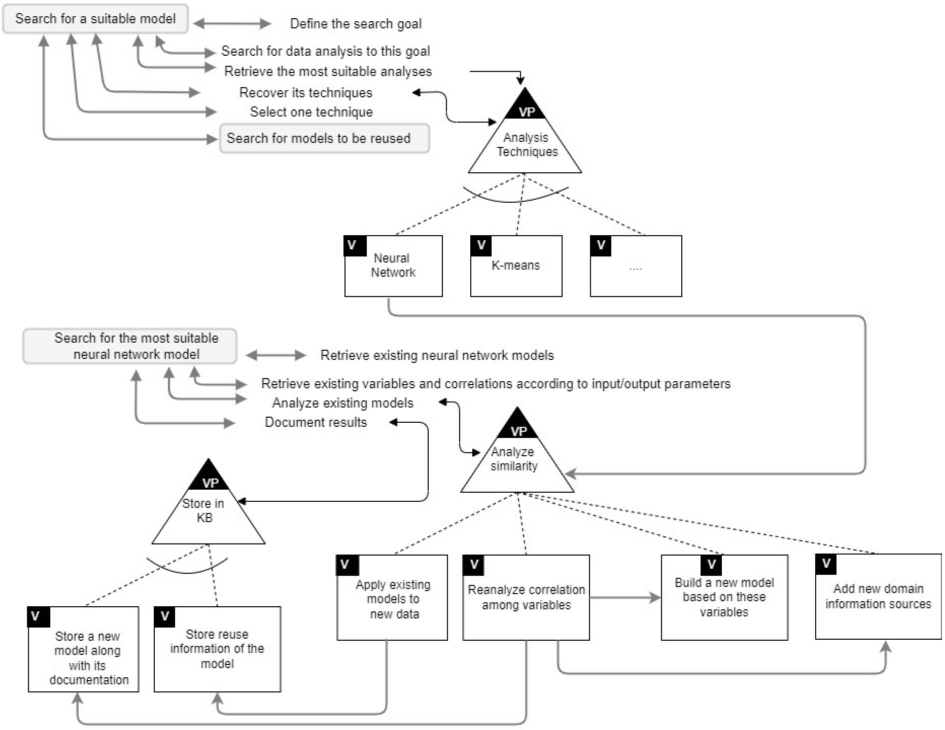

Search for a suitable model
Define the search goal
Search for data analysis to this goal
Retrieve the most suitable analyses
Recover its techniques
Select one technique
Search for models to be reused
VP
Analysis Techniques
V
Neural Network
V
K-means
V
....
Search for the most suitable neural network model
Retrieve existing neural network models
Retrieve existing variables and correlations according to input/output parameters
Analyze existing models
Document results
VP
Analyze similarity
VP
Store in KB
V
Apply existing models to new data
V
Reanalyze correlation among variables
V
Build a new model based on these variables
V
Add new domain information sources
V
Store a new model along with its documentation
V
Store reuse information of the model

Figure 1: Variety model for modeling a case of data analysis variety (extracted from [16])

way they relate each other is diverse – a DMSs might require outputs from a service by using its defined interfaces, an SPL might store a prediction provided as a result from a DMSs, and so on. Particular requirements can be seen as interacting services arranged by domain requirements, which in turn are defined and prioritized by different stakeholders.

This conceptualization is shown in Figure 2, where an AIE software system is composed of a software subsystem, such as an SPL, (left side of the figure) and a DMS subsystem (right side of the figure). The SPL is developed by following a usual Software Product Line Engineering process composed of two types of analyses [5, 6, 7]:

- Domain analysis
  Information source analysis (ISA): This activity analyzes sources of information that can support the domain analysis in order to obtain a first set of requirements.

  Subdomain analysis and conceptualization (SAC): The information recovered in the previous process is used to analyze and organize the features or services that the subdomain should offer together with the general features derived from the upper domains.

  Reusable component analysis (RCA): This process identifies the set of reusable components that could be used to implement the features defined in the last process. It returns a preliminary reference architecture.

- Organizational analysis
  Reuse and boundary analysis (RBA): This activity defines the organizational boundary, commonality, and variability features. Thus, by considering the features specified in the subdomain analysis and conceptualization process and the information from domain experts, the scope of the product line must be defined.

  Platform analysis and design (PAD): This activity builds the reference architecture based on the features defined in the previous activities and processes. The preliminary structure of reusable components defined in the reusable component analysis process is reorganized and refined. Here, each component with its common and variable parts (when necessary) is fully designed.

Datasheets are used to document variability during the Domain Analysis. As a result, we collect different ways of service compositions, which document processing requirements of the diverse functionalities of the system. These functionalities will be instantiated differently depending on the needs of a particular product. It might be an implementation for attending an operational request; or, as in our case, for providing a required processing to a DMSs. Therefore, in this last case, the services to be implemented at the left side of the figure become inputs to the DMSs Development Process (right side).

On the other way around, the DMS subsystem is developed by a DMSs Development Process, which follows the traditional activities for analytics – data

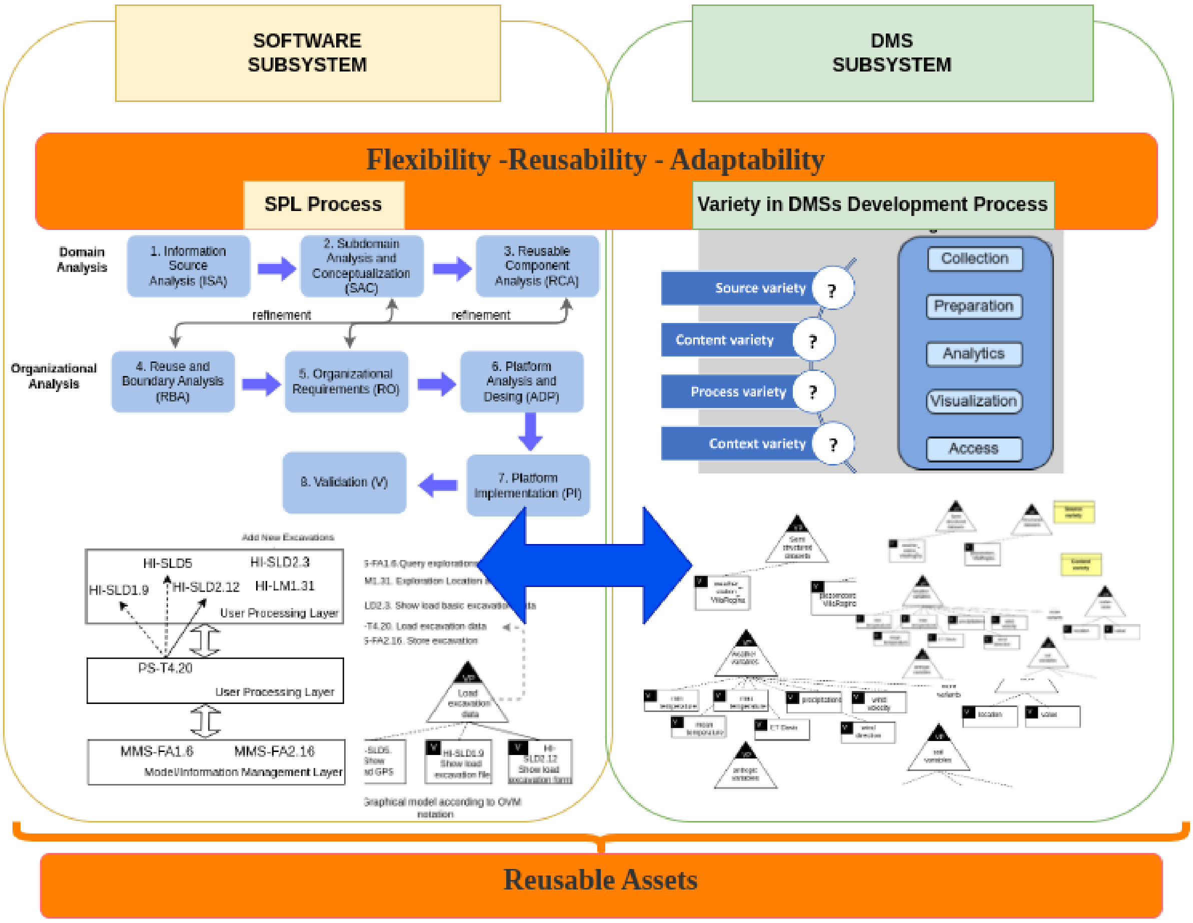

SOFTWARE SUBSYSTEM
DMS SUBSYSTEM
Flexibility -Reusability - Adaptability
SPL Process
Variety in DMSs Development Process
Domain Analysis
1. Information Source Analysis (ISA)
2. Subdomain Analysis and Conceptualization (SAC)
3. Reusable Component Analysis (RCA)
refinement
refinement
Organizational Analysis
4. Reuse and Boundary Analysis (RBA)
5. Organizational Requirements (RO)
6. Platform Analysis and Desing (ADP)
7. Platform Implementation (PI)
8. Validation (V)
Source variety
Content variety
Process variety
Context variety
?
Collection
Preparation
Analytics
Visualization
Access
Add New Excavations
HI-SLD5
HI-SLD2.3
HI-SLD1.9
HI-SLD2.12
HI-LM1.31
User Processing Layer
PS-T4.20
User Processing Layer
MMS-FA1.6
MMS-FA2.16
Model/Information Management Layer
S-FA1.6 Query explorations
M1.31. Exploration Location
LD2.3. Show load basic excavation data
-T4.20. Load excavation data
S-FA2.16. Store excavation
Load excavation data
-SLD5. Show ad GPS
HI-SLD1.9 Show load excavation file
HI-SLD2.12 Show load excavation form
Graphical model according to OVM notation
Reusable Assets

Figure 2: A conceptualization of AIE software systems

ingestion, preparation, analysis, visualization, and access. Variety influencing these activities should be detected (question marks in the figure) and documented by using our proposal of simplified datasheets. Results of this process might return as required by the SPL process and/or make a final response to decision makers.

In our proposal, we have defined top-down (T-VIP) and bottom-up (B-VIP) processes to identify variety and populate knowledge bases for DMSs, which allow store the different varieties [16]. In some way, these processes redefine domain analysis to build such repositories. For instance, meanwhile domain analysis during SPL engineering implies searching for commonalities and variabilities for the entire domain, which mirrors our top-down view, our proposal addresses domain analysis incrementally as well. The bottom-up view allows to iteratively identify variety by incorporating cases that classify variety and store it as soon as the DMS is built for one case.

The next section further discusses the elements to support DMSs Variety Management.

4.1 Managing variety in DMSs

In Figure 3 we show our model of variety management during DMSs development influenced by the four varieties discussed previously. In the top of the figure, we can find a *DMSs Development Process* component that contains the main activities (or subcomponents) of a traditional analytics.

Following, the middle of the figure shows how the variety might influence (question marks) design decisions by considering the four varieties: (1) *source variety* during *collection* helps to detect different data structures, acquisition techniques, etc., (2) *content variety* is focused on the way data should be transformed according to the business goals to be achieved, mostly considering source variations during *preparation*, (3) *process variety* helps detect variations in data analysis techniques and it is particularly involved in *analytics* and visualization tasks, and (4) *context variety* allows to identify domain variations that may constrain or affect the results of the analysis during the whole DMSs development.

These four varieties are inputs of the *Variety Management Component*. This component has been defined for answering the research question *RQ: How can we identify data variety and model reusable el-*

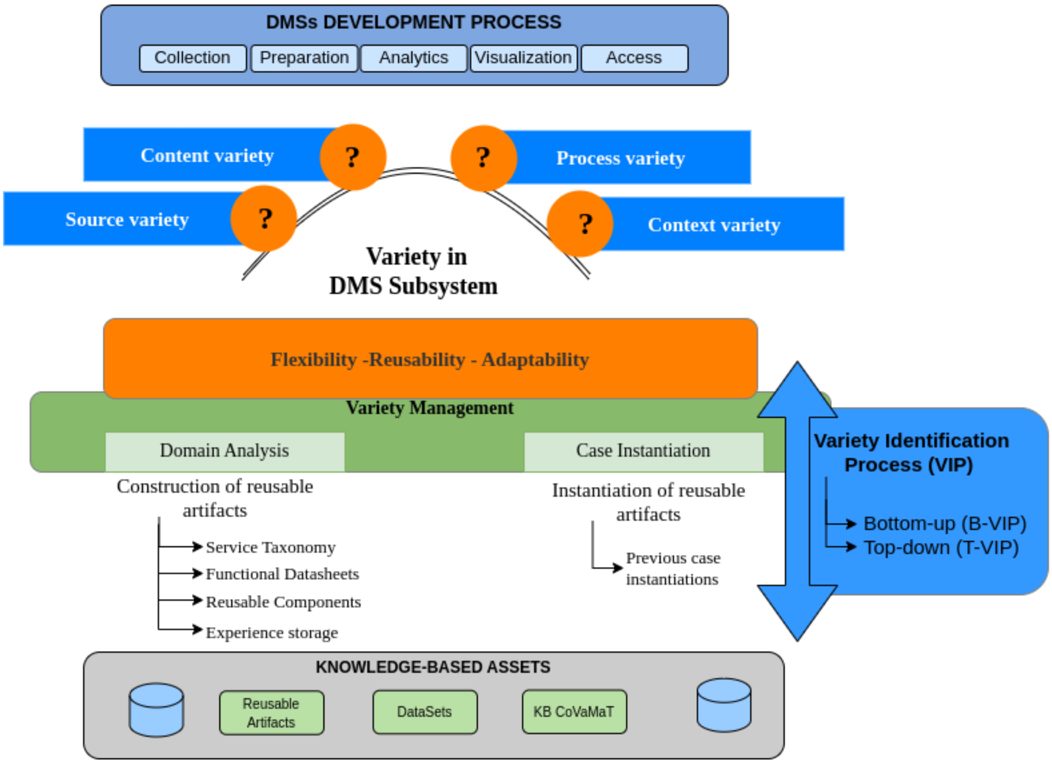


Figure 3: A model of variety management in DMSs

ements in decision-making support systems? Thus, the main quality attributes to be achieved are flexibility, reusability and adaptability by means of a *Variety Identification Process* (in blue, on the right of the figure). This process implements the two perspectives – *Bottom-up (B-VIP)* and *Top-down (T-VIP)*, which can be combined according to specific domain requirements and features [16].

At the same time, in the figure we show the definition of artifacts that are built during the adoption of these two perspectives. In the left side of the figure, *Domain Analysis* refers to the artifacts that are addressed for representing domain knowledge as taxonomies, rules given by experts, types of analytics and algorithms usually applied in this domain, etc. The right side of the figure, *Case Instantiation*, refers to the characterization of cases in terms of their varieties. It implies analyzing similarity with previous cases stored as knowledge-based assets, specifically by CoVaMaT (Context-based Variety Management Tool) [17].

The way those artifacts are identified varies depending on the approach. For example, a top-down view (T-VIP) means deeply understanding and researching domain characteristics (i.e. predicting turbidity in any water stream), probably arranging them as taxonomies (lef-side artifacts), before proceeding to contextualize and store data analysis cases (right-side artifacts) (i.e. predicting turbidity in Rio Negro river). Meanwhile, a bottom-up view (B-VIP) would start from identifying variety associated to a particular case (right side), which is stored to incrementally build domain knowledge (left-side) by adding new cases.

In addition, similarity of cases is assessed during *Case Instantiation*. This is the reusable part of the model, where varieties are compared to identify cases that might be treated similarly from an analytical point of view. For instance, predictive models for two cases might be similar when contextual features match (a motivating case in the hydrological domain can be found in [16]).

5 A case study to reuse influences on vegetation health

The case study refers requirements of the National Institute of Agricultural Technology (INTA) in its experimental station in Alto Valle of Río Negro and Neuquén[9], in the precision agriculture subdomain. Alto Valle is a region located in northern Patagonia, and surrounded by three rivers – Neuquén, Limay, and Río Negro, which delimit a productive area of approximately sixty thousand hectares, with an estimated annual production of seven hundred thousand tons of pears and apples, destined mainly for export and for the concentrated juice industry. In particular, INTA analyzes and researches areas such as plant entomology and therapeutics, plant nutrition, plant pathology, irrigation and drainage, post-harvest, crop management, etc.

In this work, we are particularly interested in the

[9] https://inta.gob.ar/altovalle/sobre-812000

spectral analysis activities to measure vegetable indices. They are defined as tools for crop development analytics, such as crop growth, vigor, biomass, chlorophyll content, etc. The vegetation indices are an indicator calculated as a result of operations with different spectral ranges of remote sensing data. The interest of these indices lies in their usefulness in the interpretation of remote sensing images; they constitute a method for the detection of land use changes (multitemporal data), the evaluation of vegetative cover density, crop discrimination and crop prediction [18]. More than 150 indices have been published in scientific literature, covering different analysis to differentiate healthy from diseased plants. For example, the Normalized Difference Vegetation Index (NDVI) describes the health of vegetation by measuring the difference between near infrared and visible red light (what vegetation reflects). Another index that we can cite is the Normalized Difference Moisture Index (NDMI) which describes the crop's water stress level. As an example of the index analyzed in this work, in Figure 4 we show the result of calculating the NDVI index in a specific zone. The figure shows two images, extracted from Sentinel-2 in August 3 and November 29, 2019, showing different values for this index. The reference value the index can take varies from -1 to 1 (from orange to green), with 1 being the healthiest possible.

5.1 Relating both subsystems: SPL-DMS

5.1.1 The software subsystem (SPL)

To populate the DMS subsystem, calculations are necessary previously at the SPL. Therefore, services that attend those calculations are modeled as functional datasheets, as we previously introduced.

Figure 5 shows the variability model for *Calculating vegetation indices*, identified by the service GS1. This functionality represents the process of generating indices from satellite imagery obtained from three satellite sources: MODIS[10], Landsat[11] and/or Sentinel-2[12]. The main differences in the images is the spatial resolution; so depending on the area that we need to analyze, we must select this resolution. The math associated with calculating a vegetation index is derived from the physics of light reflection and absorption across bands.

As a vegetation index is a number generated by combinations of remote sensing bands, for this calculus, we defined two different mechanisms – *manual* and *by using external GIS tools*. The manual method is the process of applying the calculus directly from the application, without help of any external tool; and the other mechanism is by using an external tool, such as Google Earth[13] or QGIS[14], which extracts the images and imports the vegetation index results. Finally, these results are stored as spreadsheets. The variation point representing these calculation mechanisms was defined as optional, and also, at least one variant must be selected [4]. For the last variation point (*index functions*), we left an open variation indicating that more spectral indices can be added (exemplified by NDVI, MSI, and NDMI).

5.1.2 The DMS subsystem (Variety Model)

In this case, the DMSs subsystem was implemented as a Big Data (BD) subsystem. For the analysis of vegetation indices, we firstly created a hierarchy of functionality for *analyzing factors of vegetation index*, identified as service BD2. The functionality is specific of the BD subsystem, but uses the service GS1 (Figure 5) defined in the taxonomy of the software subsystem (SPL).

Figure 6 shows the variety model of BD2, which is executed when a Big Data process is developed (left side of the figure). At the same time, all the functionality is supported by CoVaMaT through its three main functionalities. The *variety documentation* functionality (F-1) allows to specify domain varieties, which are identified through domain analysis by following a T-VIP or B-VIP process. The result of this functionality is stored in the KB as *domain variety assets*. The *case instantiation* functionality allows to instantiate a domain case, previously created by F-1, through variety instantiations. The result of this functionality is stored in the KB as *domain case assets*. Finally, the *case reuse* functionality (F-3) allows to reuse domain variety assets (created by F-1) and/or domain case assets (created by F-2). That is, domain assets previously stored can be queried and compared to a particular context aiming at detecting similar cases. When changes must be made to domain variety and/or domain case assets, a new reusable case is stored.

Specifically, BD2 defines the functionality for analyzing natural and anthropic factors that determine variation on a vegetable index (HI-BD1.1). Secondly, the call to *calculate vegetation index* functionality (GS1) is to instantiate and obtain the information about the calculus of the specific index or indices to be analyze here. The following two services, *Search for similar objectives/cases* (GS5) and *Retrieve existing domain/-case/reusable assets* (GS6), are defined for searching and recovering similar domain and/or case applications in CoVaMaT KB (by F-3). This functionality allows stakeholders to reuse previously created assets (if exist). If they do not exist, that is, there is no domain or case application stored in the KB, we define a service for *analyzing similarities to be instantiated/reused* (PS-SS3). Notice that reusing does not imply exact

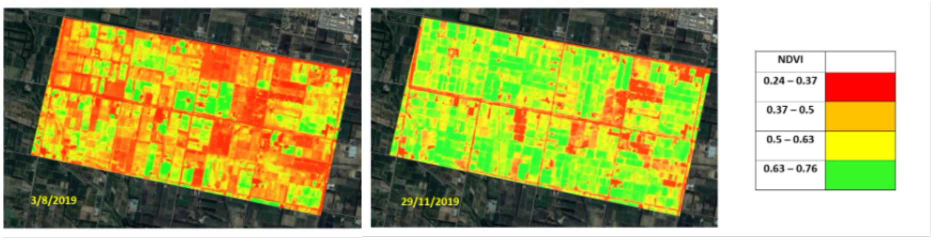
[10] `https://modis.gsfc.nasa.gov/`

[11] `https://landsat.gsfc.nasa.gov/`

[12] `https://sentinel.esa.int/web/sentinel/missions/sentinel-2`

[13] `https://earth.google.com/web/`

[14] `https://www.qgis.org/`



a) NDVI image extracted on 3 August 2019

b) NDVI image extracted on 29 November 2019

Figure 4: NDVI images extracted from Sentinel-2 of Alto Valle region on August and November,2019

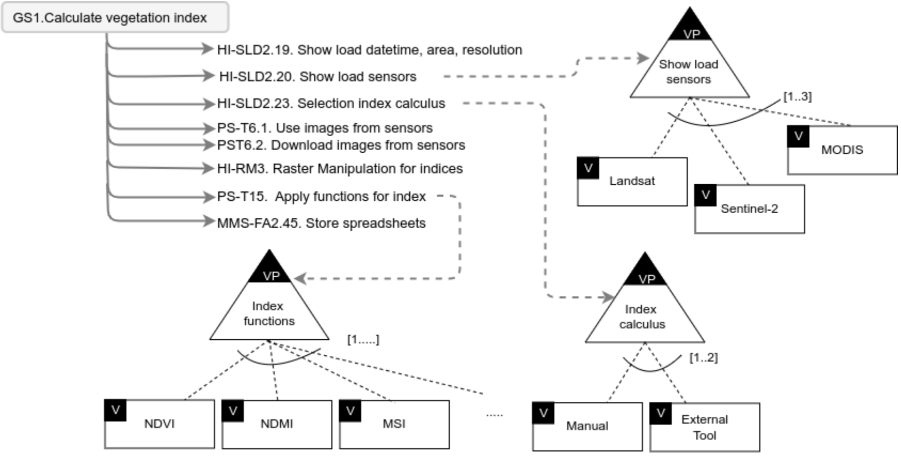

GS1.Calculate vegetation index
HI-SLD2.19. Show load datetime, area, resolution
HI-SLD2.20. Show load sensors
HI-SLD2.23. Selection index calculus
PS-T6.1. Use images from sensors
PST6.2. Download images from sensors
HI-RM3. Raster Manipulation for indices
PS-T15. Apply functions for index
MMS-FA2.45. Store spreadsheets
VP
Show load sensors
[1..3]
V
MODIS
V
Landsat
V
Sentinel-2
VP
Index functions
[1.....]
V
NDVI
V
NDMI
V
MSI
.....
VP
Index calculus
[1..2]
V
Manual
V
External Tool

Figure 5: Variability model (SPL) for *calculating vegetation indices*

matching of domain assets. Thus, the stored assets can be retrieved and compared to a particular context aiming at detecting similar cases. In case the information cannot be reused at all, we define a variation point (*Define cases*) in order to create these specific assets (variety/domain/reusable assets).

Following, the *Find/Add information sources* service (MMS-FA2.53) retrieves and/or stores data sources needed for the analysis activities. The datasets are not stored in CoVaMat KB, they are part of another repository instead. These datasets must be then pre-processed to be adapted for the application case (PS-BD4), and the analysis must be re-made (PS-SS4 and PS-BD5).

Finally, all the defined variations, data sources, and results are stored in the KB (GS7). To do so, the F-1, F-2 and F-3 functionalities can be executed by defining another variation point, called *Store in KB*. This point is defined as an optional one, and restricted to select at least one variant. The variants allow to store domain/-case assets, by *Store a new domain model along with its documentation*) variant, or reusable assets by the *Store reuse information of the analyses* variant.

5.2 Defining and instantiating the case

In order to show how the instantiations of the two subsystems are performed, we describe the case study with its specific requirement, *determining factors affecting the NDVI index.*

Software subsystem instantiation

For the specific requirement, we firstly instantiated the GS1 functionality (Figure 5) in order to use the Sentinel-2 sensor and an external tool for NDVI calculus. In Figure 7 we can see the instantiations, in which the result was an spreadsheet with areas and NDVI values from 2015 to 2022.

DMS instantiation

For the DMS instantiation, we describe our case study for the construction of two Big Data Subsystems (BDSs). To do so, we performed three main activities: (1) we applied a bottom-up approach (B-VIP) for variety identification, that is, we defined the configuration of each variety for the two BDSs; (2) we instantiated the *BD2. Analyze factors of vegetation index* variety model (Figure 6) to analyze factors in a specific area; and (3) we instantiated again BD2 to analyze the same factors in another area. Each of these activities are described as follows.

(1) B-VIP: Variety identification

We firstly applied a bottom-up approach (B-VIP), starting from user's requirements, and performing an

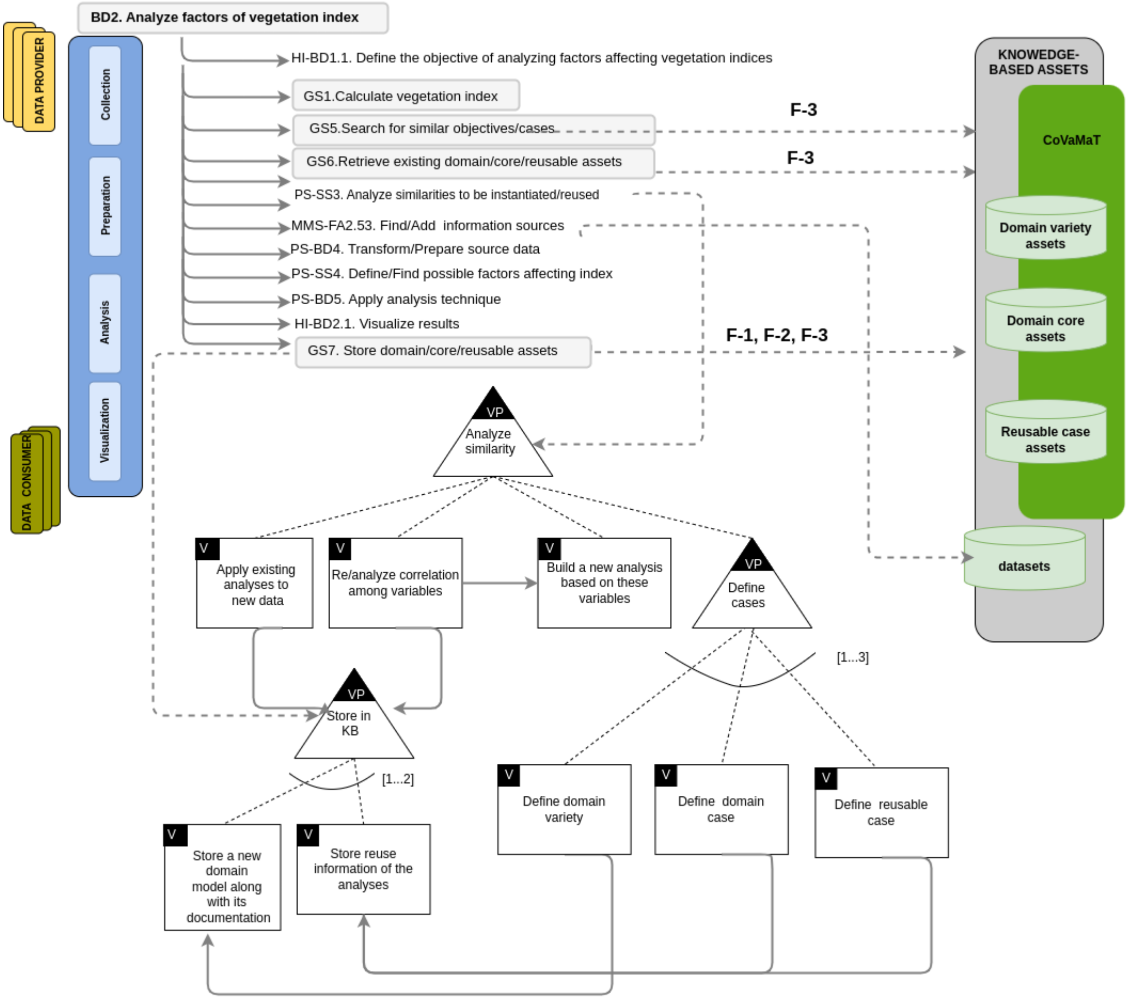

BD2. Analyze factors of vegetation index
DATA PROVIDER
Collection
Preparation
Analysis
Visualization
DATA CONSUMER
HI-BD1.1. Define the objective of analyzing factors affecting vegetation indices
GS1.Calculate vegetation index
GS5.Search for similar objectives/cases
GS6.Retrieve existing domain/core/reusable assets
PS-SS3. Analyze similarities to be instantiated/reused
MMS-FA2.53. Find/Add information sources
PS-BD4. Transform/Prepare source data
PS-SS4. Define/Find possible factors affecting index
PS-BD5. Apply analysis technique
HI-BD2.1. Visualize results
GS7. Store domain/core/reusable assets
F-3
F-3
F-1, F-2, F-3
KNOWEDGE-BASED ASSETS
CoVaMaT
Domain variety assets
Domain core assets
Reusable case assets
datasets
VP
Analyze similarity
V
Apply existing analyses to new data
V
Re/analyze correlation among variables
V
Build a new analysis based on these variables
VP
Define cases
[1...3]
VP
Store in KB
[1...2]
V
Define domain variety
V
Define domain case
V
Define reusable case
V
Store a new domain model along with its documentation
V
Store reuse information of the analyses

Figure 6: Variety model for *analyzing factors of vegetation index*

exploratory study in order to analyze if they can be satisfied.

For the same requirement (*determining factors affecting the NDVI index*), varieties were defined generating the construction of two BDSs. In particular, we defined a case study of *context variety* trying to answer how weather and a water table (measures from piezometers[15]) may affect design decisions. In order to avoid influences from the other types of variety, sources, content, and processing varieties remain stable. That is, as Figure 8 shows, we collected data from the same sources and with the same format for two areas, *Area1* and *Area2*, containing different NDVI indices; transformed and prepared data in the same way for the two areas; and used the same analysis techniques in both cases. In this way, we expected to reduce effects of additional variations on the case study. Specifically, varieties were defined as follows:

- *Source variety* of BD System1-BD System 2: involved the same sources. The Alto Valle region has several weather stations located in different geographic points, which provide information for weather forecasts. In particular, in Villa Regina subregion, there is one station (*Villa Regina weather station* located at X:663742, Y:5666849)[16] registering information each ten minutes. It includes temperature, atmospheric pressure, humidity, wind speed, wind direction, precipitations, and evapotranspiration (ET). The data generated are semi-structured in text format. Another source considered here was the water table. For this table INTA has installed more than fifty piezometers in Villa Regina subregion. Data are collected manually once a month and registered in a structured file (as a spreadsheet file).

- *Content variety* of BD System1-BD System 2: involved the same variables. Specifically, the content involved in these systems includes, from weather stations, temperature (min, max and mean), precipitations (in mm), wind velocity

[15] Pressure-sensitive, submersible measurement sensors to detect water pressure and levels.

[16] Coordinate system WGS 84 UTM ZONE 19S

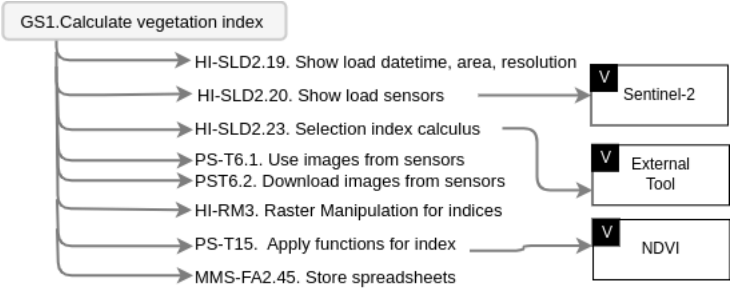

GS1.Calculate vegetation index
HI-SLD2.19. Show load datetime, area, resolution
HI-SLD2.20. Show load sensors
HI-SLD2.23. Selection index calculus
PS-T6.1. Use images from sensors
PST6.2. Download images from sensors
HI-RM3. Raster Manipulation for indices
PS-T15. Apply functions for index
MMS-FA2.45. Store spreadsheets
V
Sentinel-2
V
External Tool
V
NDVI

Figure 7: Instantiated variability model (SPL) for *calculating NDVI index*

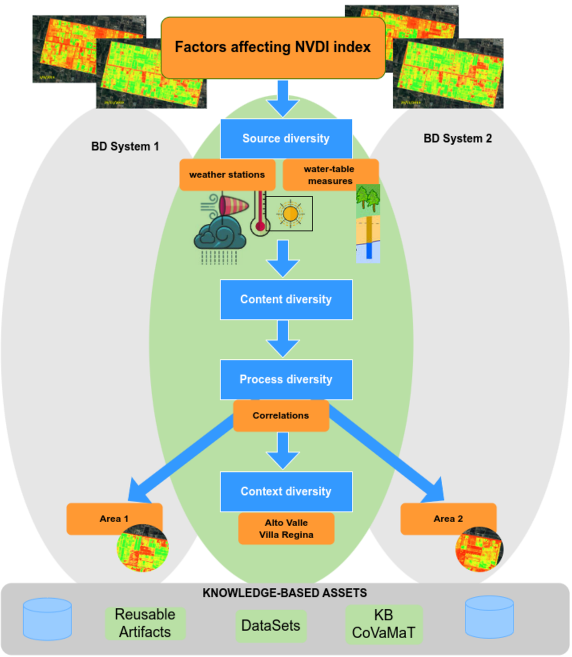


Figure 8: Two BD subsystems generated by context variety

(mean and max), wind direction, humidity (min, max and mean), global radiation (max and mean), and ET. For the piezometers, data were collected once a month.The period considered for populating the sources was 2015-2022.

- *Process variety* of BD System1-BD System 2: involved the same processing type and data analysis techniques. We performed a batch processing in which source data was processed offline. Following, for the data analysis techniques, we applied the same type of correlation analysis (Pearson) for the two systems.

- *Context variety*: here we had the information of the NDVI index, whose measures were extracted and calculated from images of Sentinel-2 sensors, once a month from 2015 to 2022. Each measure covers an area of 10x10 meters (100 m2). The context was analyzed by extracting two areas (*Area1* and *Area2*) with different NDVI values, aiming at determining reasons for this variation from analyzing weather and water table data. The areas were selected considering the proximity to the Rio Negro River, and the proximity to the plateau edge surrounding the valley. The map in the top of Figure 14 shows their precise locations.

  – *BD System1*: Area 1, near the river (centroid X:2665039.06, Y:5664026.04)

  – *BD System2*: Area 2, near the plateau edge (centroid X:2675630.46, Y:5668533.27)

(2) System1-BD

Once the varieties were identified, we instantiated the BD2 (Figure 6) considering we had no information stored in CoVaMaT. Figure 9 shows this instantiation in which we used the output of GS1 (information of NDVI indices in Villa Regina region) and searched for similar domain cases in CoVaMAT (F-3). As we did not have variety/case assets stored, we instantiated the *define domain case* and *define domain variety* variants in order to create *domain variety* and *domain case* assets (as we defined in variety identification previously - B-VIP).

The *domain variety* asset was developed by considering the sources available and the information provided by expert users. Thus, in Figure 10 we can see the four varieties (in yellow) containing the options (domain variants) that each of them can contain for the requirement. For example, *source variety* includes the sources obtained from weather stations, piezometers, etc.; *content variety* includes information about possible factors that can affect the NDVI index, such as weather variables, population, soil types, etc.; *process variety* includes the type of processing (batch, real-time, etc.) and the type of analysis (in this case a correlation); and the *context variety* only defined for a specific area (*Area1*). All these varieties were stored in CoVaMaT as *domain variety 1*.

Also, as we were developing an application case (and not only defining the domain varieties), we developed the *domain case 1* containing instantiations of the variants defined in *domain variety 1*). Figure 11 shows these instantiations where the configurations

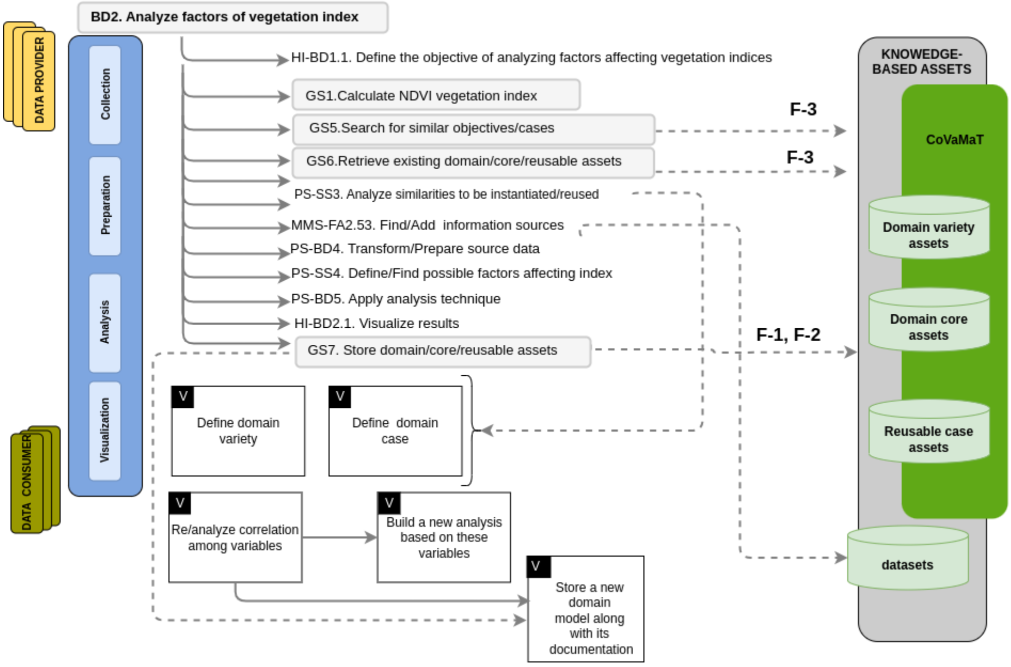

BD2. Analyze factors of vegetation index
DATA PROVIDER
Collection
Preparation
Analysis
Visualization
DATA CONSUMER
HI-BD1.1. Define the objective of analyzing factors affecting vegetation indices
GS1.Calculate NDVI vegetation index
GS5.Search for similar objectives/cases
GS6.Retrieve existing domain/core/reusable assets
PS-SS3. Analyze similarities to be instantiated/reused
MMS-FA2.53. Find/Add information sources
PS-BD4. Transform/Prepare source data
PS-SS4. Define/Find possible factors affecting index
PS-BD5. Apply analysis technique
HI-BD2.1. Visualize results
GS7. Store domain/core/reusable assets
F-3
F-3
F-1, F-2
KNOWEDGE-BASED ASSETS
CoVaMaT
Domain variety assets
Domain core assets
Reusable case assets
datasets
V
Define domain variety
V
Define domain case
V
Re/analyze correlation among variables
V
Build a new analysis based on these variables
V
Store a new domain model along with its documentation

Figure 9: System1-BD: Instantiated variety model for *analyzing factors of NDVI index*

described for *BD System1* in the B-VIP process were applied.

Thus, the sources were a set of climate variables provided by the Villa Regina weather station, and the measures of piezometers manually obtained once a month. All of these sources included the 2015-2022 period. Then, the measures of wind velocity and direction, max, min and media temperatures, precipitation and evapotranspiration, were selected from the content; and a batch processing and correlation analysis were instantiated from process variety. The result was a new domain case, stored as *domain case 1*.

Following, we applied the configurations of this domain case (Figure 9), by performing the services PS-BD4 (for processing and prepare source data), PS-SS4 (for analyzing factors), and PS-BD5 (for applying the correlation analysis). For this analysis, and as there was no previous analysis stored, we instantiated *re/analyze correlation among variables* and *build a new analysis based on these variables* variants in BD2.

Finally, we stored the *domain variety 1* and *domain case 1* assets and in the KB (GS7) by using F-1 and F-2 functionalities of CoVaMaT.

(3) System2-BD

To highlight the importance of defining context variety to promote reusability, here we developed a new system to analyze the same factors but in a diiferent context, in this case is a different area (*Area 2*). In this case, the BD2 variety model (Figure 6) was instantiated in a different way but considering we had information about the same objective in the KB. Thus, in Figure 12 we can see a set of different variants selected. In particular, by using F-3 functionality, we could retrieve *domain variety 1* and *domain case 1* assets, and therefore we instantiated the *Define reusable case* variant.

In this case, once the *domain variety 1* was retrieved, we realized that we needed a new variant for the context definition, because only *Area 1* was previously defined. Thus, we were facing a *context variety case*, and we had to create a new variant, named *Area 2*. Therefore, the domain variety asset was changed to contain two variants, *Area* 1 and *Area 2* (Figure 13a)). Following, we searched for a similar domain case asset, and the *domain case 1* was retrieved. Here, we again analyzed the instantiations in order to see if we needed a different configuration (PS-SS3 service of BD2). In this case, we had only to change the instantiation of the context variety by choosing the *Area 2* (Figure 13b)). Finally, the *reusable case 2* was stored in KB (GS7 service).

Results of the two BD Systems

In Figure 14 we show the results of executing both processes (Figure 9 and 12).

Firstly, significant correlation (grater than 0.5) showed that piezometer's values do not influence NDVI values in these areas. As we can see from Figure 14, temperature (min, max and mean), global radiation (max and mean), and ET were the most significant influences. However, their intensities vary – *Area1* shows stronger correlation (around 80%) than *Area2* (around 60%).

In *Area1* (in green), NDVI values trend upwards indicating healthier vegetation (as the scatter plot shows);

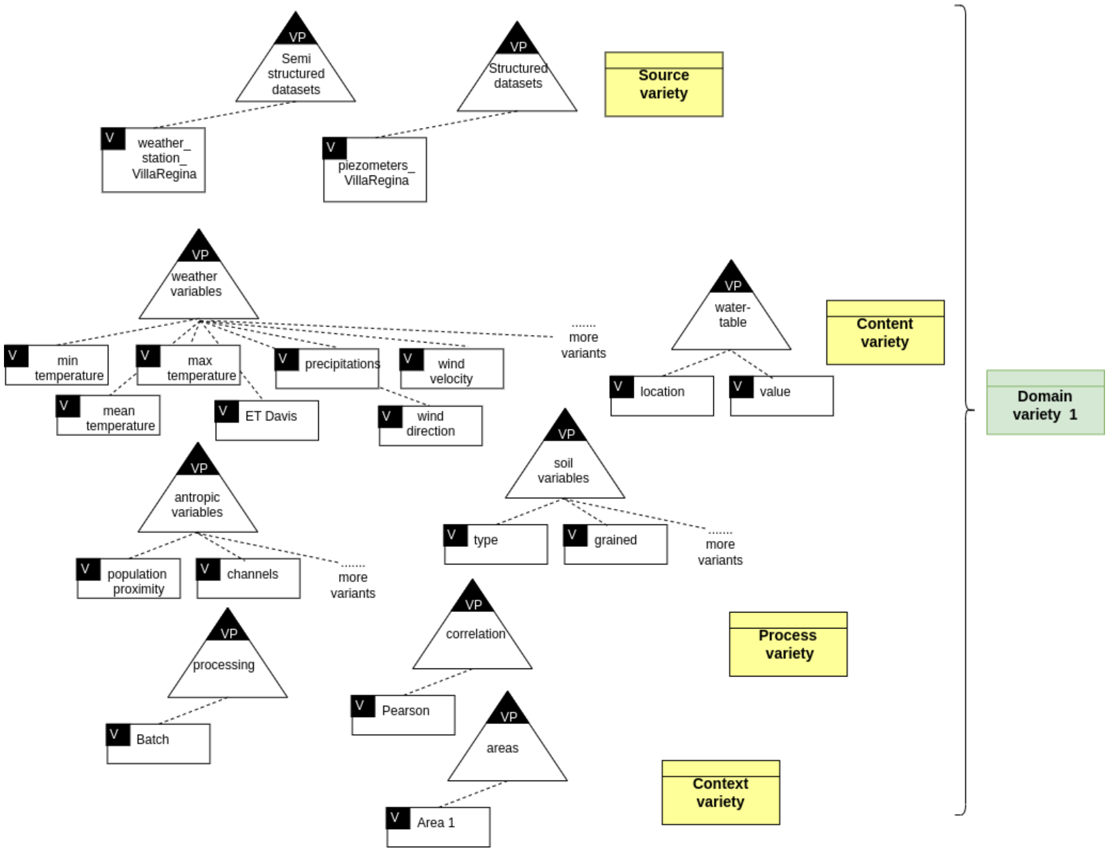


Figure 10: Definition of the domain variety

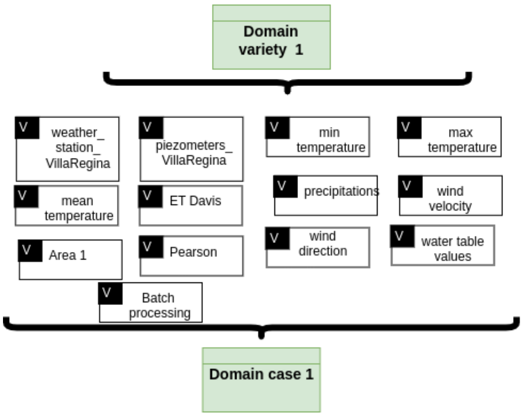


Figure 11: Definition of the domain case

and stronger correlations indicate that these factors are more significant near the river. On the contrary, correlations in *Area2* are less significant and NDVI values trend downwards (in red).

These results were shown to the experts users to validate the analysis. Causes are now the point to be analyzed. For instance, temperature might be more significant near a water stream due to the influence of a water body, which usually makes the temperature values not so extreme. It may cause smoother conditions for vegetation grow; so, great variations of temperature may cause strong effects on vegetation's health. Of course, all of this depends on the type of vegetation. This factor was not included in the current version of the analysis; however, as an illustrative case, it shows the contextual reuse possibilities (an hypothetical *Area3*, might reuse previous analyses closer to its contextual characterization).

6 Conclusion and Future Work

In this paper, we have introduced a proposal for modeling reusable assets from an integrated view of AIE software systems. This integration relies on using the conceptualization of variability from a software product line development along with our approach for variety management in decision-making support system development. A case study, as a proof of concept, showed the applicability; however there is still much work to do. Firstly, the supporting tool (CoVaMaT) is a prototype that needs further development; secondly, empirical validation trough case studies are continuously populating CoVaMaT in cooperation with domain experts at INTA Alto Valle; and finally, some extensions of the model to include domain standards are in progress.

Competing interests

The authors have declared that no competing interests exist.

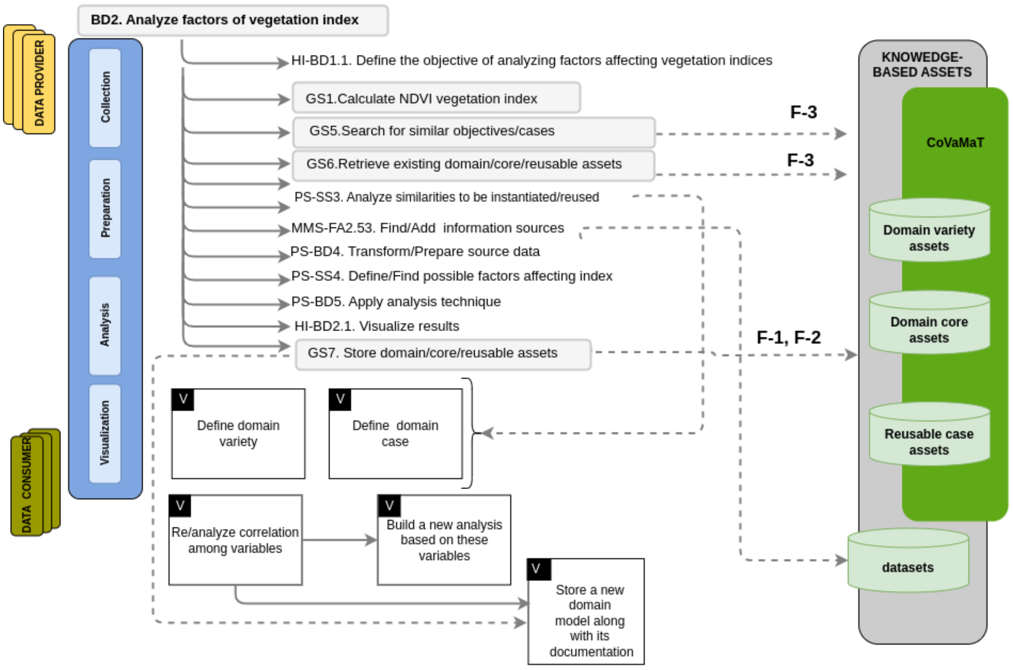

BD2. Analyze factors of vegetation index
DATA PROVIDER
Collection
Preparation
Analysis
Visualization
DATA CONSUMER
HI-BD1.1. Define the objective of analyzing factors affecting vegetation indices
GS1.Calculate NDVI vegetation index
GS5.Search for similar objectives/cases
GS6.Retrieve existing domain/core/reusable assets
PS-SS3. Analyze similarities to be instantiated/reused
MMS-FA2.53. Find/Add information sources
PS-BD4. Transform/Prepare source data
PS-SS4. Define/Find possible factors affecting index
PS-BD5. Apply analysis technique
HI-BD2.1. Visualize results
GS7. Store domain/core/reusable assets
F-3
F-3
F-1, F-2
V
Define domain variety
V
Define domain case
V
Re/analyze correlation among variables
V
Build a new analysis based on these variables
V
Store a new domain model along with its documentation
KNOWEDGE-BASED ASSETS
CoVaMaT
Domain variety assets
Domain core assets
Reusable case assets
datasets

Figure 12: System2-BD: Instantiated variety model for *analyzing factors of NDVI index*

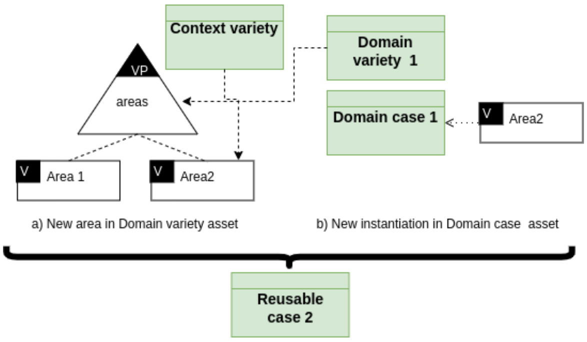


Figure 13: Definition of a reusable case

Authors' contribution

The first two authors contributed equally to this research. They discussed the results and contributed to the final manuscript. Both authors read and approved the final manuscript.

Carolina Villegas contributed in the DMS instantiation activity through the execution of the instantiated variety model and the visualization of results. Finally, INTA staff contributed as expert users providing also source data, knowledge and analysis of the results.

All authors read and approved the final manuscript.

Funding

This work is partially supported by Universidad Nacional del Comahue, Project 04/F019-Modelado de Variedad en Sistemas Big Data

References

[1] J. Bosch, H. H. Olsson, B. Brinne, and I. Crnkovic, "Ai engineering: Realizing the potential of ai," *IEEE Software*, vol. 39, no. 6, pp. 23–27, 2022.

[2] P. C. Clements and L. Northrop, *Software Product Lines : Practices and Patterns*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[3] F. van der Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[4] K. Pohl, G. Böckle, and F. J. v. d. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[5] A. Buccella, A. Cechich, M. Arias, M. Pol'la, S. Doldan, and E. Morsan, "Towards systematic software reuse of gis: Insights from a case study," *Computers & Geosciences*, vol. 54, no. 0, pp. 9 – 20, 2013.

[6] A. Buccella, A. Cechich, M. Pol'la, M. Arias, S. Doldan, and E. Morsan, "Marine ecology service reuse through taxonomy-oriented SPL development," *Computers & Geosciences*, vol. 73, no. 0, pp. 108 – 121, 2014.

[7] A. Buccella, A. Cechich, J. Porfiri, and D. Diniz Dos Santos, "Taxonomy-oriented domain analysis of gis: A case study for paleontological software systems," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, 2019.

[8] B. Custers and H. Uršič, "Big data and data reuse: a taxonomy of data reuse for balancing big data benefits and personal data protection," *International Data Privacy Law*, vol. 6, no. 1, pp. 4–15, 2016.

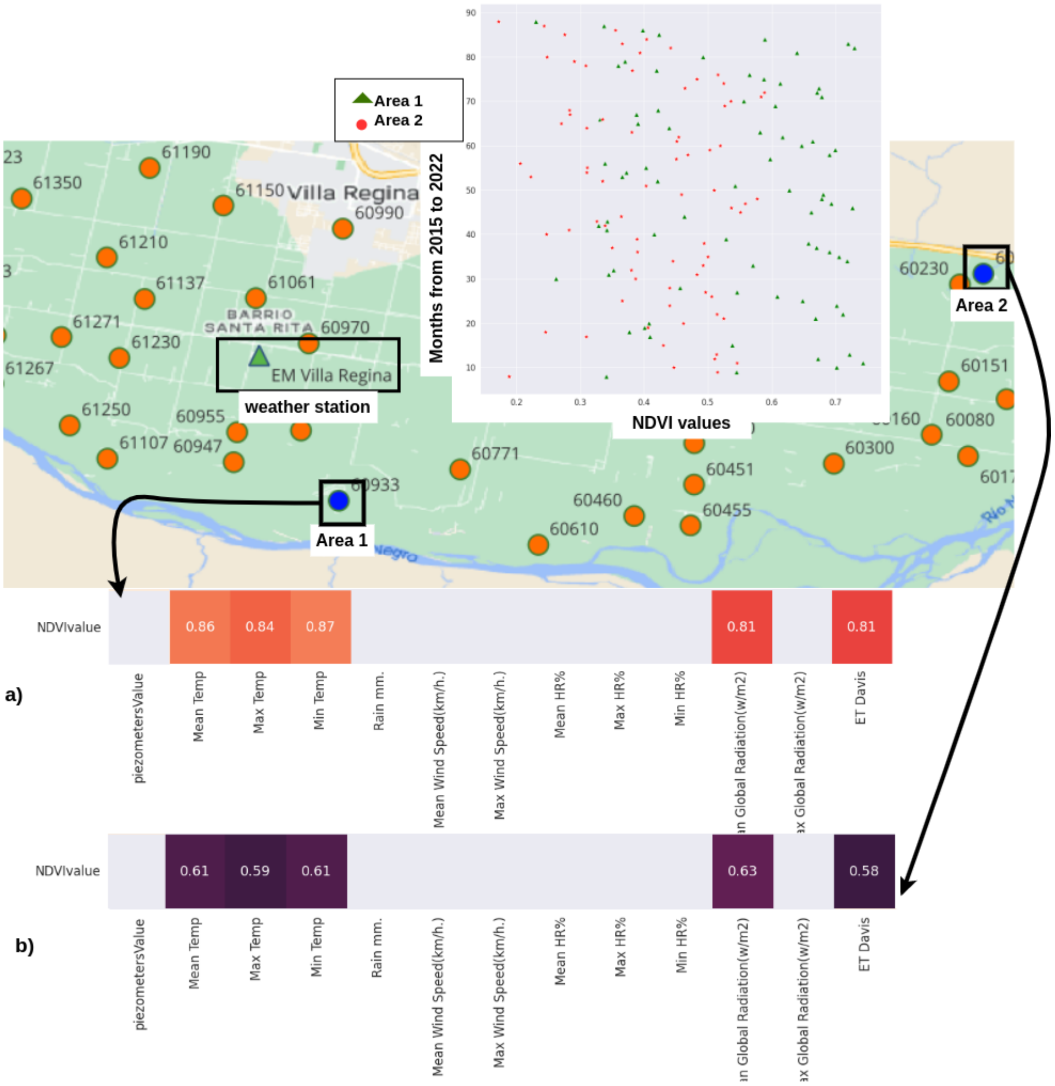


Figure 14: Relation between NDVI index and climate and water-table variables in the two areas, when the correlation is more than 0,5

[9] I. Pasquetto, B. Randles, and C. Borgman, “On the reuse of scientific data,” *Data Science Journal*, vol. 16, no. 8, 201720.

[10] Z. Xie, Y. Chen, J. Speer, T. Walters, P. A. Tarazaga, and M. Kasarda, “Towards use and reuse driven big data management,” in *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries*, p. 65–74, Association for Computing Machinery, 2015.

[11] R. Borrison, B. Klöpper, M. Chioua, M. Dix, and B. Sprick, “Reusable big data system for industrial data mining - a case study on anomaly detection in chemical plants,” in *Intelligent Data Engineering and Automated Learning – IDEAL 2018*, pp. 611–622, Springer International Publishing, 2018.

[12] W. Epperson, A. Yi Wang, R. DeLine, and S. M. Drucker, “Strategies for reuse and sharing among data scientists in software teams,” in *Proceedings of ICSE-SEIP ’22, Pittsburgh, PA, USA*, Association for Computing Machinery, 2022.

[13] J. Klein, “Reference architectures for big data systems, carnegie mellon university’s software engineering institute blog.” http://insights.sei.cmu.edu/blog/reference-architectures-for-big-data-systems/ (Accessed June 9, 2021), 2017.

[14] M. Pol’la, A. Buccella, and A. Cechich, “Analysis of variability models: a systematic literature review,” *Softw. Syst. Model.*, vol. 20, pp. 1043–1077, 2020.

[15] J. Abawajy, “Comprehensive analysis of big data variety landscape,” *International Journal of Parallel, Emergent and Distributed Systems*, vol. 30, no. 1, pp. 5–14, 2015.

[16] L. Osycka, A. Buccella, and N. A. Cechich, “Data variety modeling: A case of contextual diversity identification from a bottom-up perspective,” in *27th Argentine Congress, CACIC 2021, Salta, Argentina, October 4-8, 2021, Revised Selected Papers. Communications in Computer and Information Science 1584*, pp. 124–138, Springer, 2022.

[17] L. Osycka, A. Cechich, A. Buccella, A. Montenegro, and A. Muñoz, “Covamat: Functionality for variety reuse through a supporting tool,” in *XI Conference on Cloud Computing, Big Data & Emerging Topics (JCC-BD&ET)*, 2023.

[18] B. Abderrazak, D. Morin, F. Bonn, and A. Huete, “A review of vegetation indices,” *Remote Sensing Reviews*, vol. 13, pp. 95–120, 01 1996.

Citation: A. Cechich, A. Buccella, C. Villegas, A. Montenegro, A. Muñoz, A. Rodriguez. *A Model of Reusable Assets in AIE Software Systems.* Journal of Computer Science & Technology, vol. 23, no. 2, pp. 145–160, 2023.
DOI: 10.24215/16666038.23.e13
Received: April 4, 2023 **Accepted:** July 14, 2023.