# Using Ontological Methods for Managing Product Development

Henson Graves
*Algos Associates, 2829 West Cantey Street, Forth Worth, MyCountry*
*henson.graves@hotmail.com*

Keywords: Conceptual Modeling, Foundation ontologies, Metadata, Ontology, Product Data Management, Product Development.

Abstract: Managing product development requires the ability to plan the work to be accomplished and the ability to evaluate whether the work has been accomplished successfully. Managing product development increasingly depends on producing and evaluating digital work products. The information needed involves a wide variety of subjects: products and their components, manufacturing and test equipment, organizations, physical processes, product requirements and designs, as well as analysis, test, and status results. A product development information system which uses a conceptual model (ontology) as the bais for organizing these subjects can assess whether the digital work products needed for program milestones are available and meet their acceptance criteria. The ontology can be used to produce specifications for these digital work products that can be used interactively with an information system for tasks such as validation and generating discrepancy reports. A product development ontology replaces traditional lists of documents with their metadata to give a much more accurate and accessible information picture of a program. A product development ontology does not have to be constructed from ground zero, it can be constructed by extending a foundation ontology with domain knowledge concerning product development.

## 1 INTRODUCTION

Product development refers to the lifecycle of a manufactured product from early conceptualization through design, manufacture, verification, delivery, operation, and final disposal. As product development has evolved from putting together physical prototypes, to use of extensive paper document libraries for program documents, to digital data, the issue of information accuracy, pedigree and provenance becomes acute. Invalid and missing information is a primary cause of cost, schedule overrun, and redesign (Graves et al., 2004). Managing product development relies increasingly on the production and evaluation of digital artifacts. These artifacts have become the authoritative source information about actual products development. When these are flawed and incomplete, determining and predicting the status of a program toward meeting its goals and milestones is difficult, if not impossible.

Program planning requires the capability to describe the tasks to be accomplished and the work products to be produced. How to sequence tasks, what the resulting work products are, and how they are to be evaluated constitute the enterprise knowledge base. Program assessment requires answering questions such as: what data is required for milestone event completion, what data has actually been produced, and does the data satisfy its requirements. To assess progress, one must be able to discover and access the digital work products and check that they have sufficient quality, as well as check that the entire collection is consistent. Answering these questions with confidence requires specifications for the digital artifacts can be verified and that their representation reflect the actual physical world. Even when the artifacts exist a consumer may not be aware of their existence, and may not be able locate them.

This work focuses on issues of what information is needed to quantitatively determine whether a program development program is on track to meet its milestones. It complements and overlaps the development of ontologies for collaborative product development that focus on design representation (Borst, 1997), assembly information (Kim et al., 2006), product life cycle management (Sudarsan et al., 2005), and with ontologies used in the description and operation of a product (Graves, 2007). A product development ontology of this kind models physical reality including products and their components, and events, as well

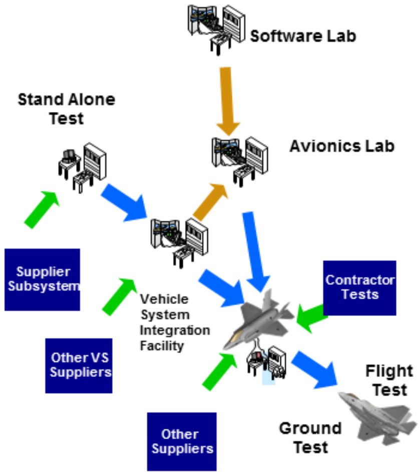


Figure 1: Information Flow leading to first flight.

as, work products (information objects) such as plans and the work products called out by the plans. An approach to determine the categories for the subject of product development and their relationships and properties needed for assessment is outlined.

Figure 1 illustrates an example of information flow and task dependences leading to the first flight of the first article of an aircraft model. At each stage of the process there is documentation of the outcomes of equipment and software tests. The different test labs are configured with software models of the context of the component under test. The tests are often iterated until satisfactory results are achieved. However, determining if all of the prerequisite criteria have been met is an imposing task. For the prerequisite tasks to have been successfully accomplished requires more than a report saying that the task has been accomplished. It requires the ability to probe into, for example, what software models where used in a component testing configuration. Answering this level of detail stresses the capability of conventional information systems.

How can the information content of work products be described in a way that stakeholders can recognize, identify, and assess its content? Where the number of work products is small, on the order of a few hundred, this task can be managed without complex information systems. Where the product has hundreds of subsystems made up of thousands of assemblies and parts, and each component requires the production of many work products, such as requirements, design, and test conditions, an information system is required to enable identification, access, discovery, and assessment of all of the work products relevant to a task (Graves et al., 2004). The inability to know that the right work products are being produced and to know that the work products have the right content and quality is a major contributor to program cost, schedule, and integrity issues. It is not enough to be able to answer the questions; we need to have confidence that the answers are correct. Correct answers require understanding of relationships between work products, such as what their components are, and how a work product uses other data as input for its creation (Graves, 2007).

In the same sense that product specifications determine what is to be built and serve as criteria to recognize whether something satisfies the product specification, a digital work product specification defines the information content of the work products needed for a development program identifies the work product evaluation criteria. Only when work products have predefined specifications and actual work products are assessed against their specifications can the planning for a program to be effective and accurate assessments of program execution be made. Of course assessment can only be made on the basis of work product specifications and assessment criteria.

Information storage and retrieval systems based on metadata provide a solution to the a very important problem, the fact that many design errors trace to superseded data being used in the production of new data. Metadata can provide data pedigree, but doesnt in itself model data content and its relation to physical products being designed and manufactured or the components of the enterprise used in product development (Scrudder et al., 2003).

The stakeholders in a development process each have some informal conceptual model of the enterprise and its workings. The stake holder's models may be somewhat different and may only have depth in specific areas. A product development ontology repesents a program level specification for what digital work products are to be developed and how they are to be assessed. The ontology defines a model of the program enterprise, the digital work products, and specifies how the digital work products are connected to the actual physical reality. The ability of the ontology to make the connection between the digital world and the physical world is the key to quantitative program assessment.

The idea of developing a conceptual model (ontology)for the enterprise and configuring an information system to use it extends more traditional use of information models to configure information systems. Constructing an ontology requires analysis of needs of planners, producers, consumers, and assessors; the construction also reflects experience with issues of identification, and development of work product descriptive content sufficient for the stakeholder needs. Several authors have advocated developing an ontology for use by an information system to provide not only traditional configuration management control, but also to provide value-added services (Lagoze and Hunter, 2001).

An information system can use the ontology to provide detailed views into the state of program execution as compared with planned execution. By using an ontology metadata for discovery and access is that the ontology can answer what data is required and provide specifications for the data. For example, a workflow, identifies tasks with input and output work products, as described by their specifications. The task occurrence is characterized by inputs, outputs, resources used to execute the task, as well as when it occurred. A task execution record is an information object that provides a view onto the task occurrence. If work product classes have definitions, then the definitions can be used to generate Web-forms for data input. Filling in the form creates a validated instance of the class represented by the form. For example a work flow for flight certification identifies tasks that lead to flight certification. Each task in the work flow specifies the work products that are output and the criteria that are used for verifying the work products for task completion. The result is a tree of certification reports for each of the integration tasks. The final certification report will contain restrictions on the flight envelope and usage life of the component parts.

The next section will illustrate the task of developing work product specifications by considering what work products are required for an air frame to be certified and which ones have been produced.

# 2 DIGITAL INFORMATION PRODUCTS

Work products are the digital artifacts produced as a result of program tasks. Work products include plans and workflows, as well as, requirements, specifications, test conditions, and test results that the plans entail. Plans are a good place to start in attempting to understand the scope of product development subjects. If a digital work product is actually specified to be produced it will be indirectly referenced through a chain of work products from a program plan. To develop a product development ontology plans are a good place to start.

Large product development enterprises generally have lists of plans that a product development program is expected to produce. The list may be tailored for individual program needs. Generaly there are examples of what these plans are expected to contain and in some cases there are plan templates. However, there may not be detailed specifications of the data content and maturity needed to accomplish a program task and allow potential consumers to determine if a work product that meets it specification is what they need. For example, an analysis report specifies what characteristics of a product have been measured and what methods were used to obtain the results. A physical description of a product may specify structural layout and component positioning as part of the design of the product.

**Plan**

- Applies To: safety of all product variants
- hasGoal: meet system safety requirements
- hasSuccessCriteria: contract requirements, industry safety
- hasRole: flight test safety process, verification process
- definesTask: hazard analysis, identification of safety critical functions
- usesMethod: fault tree analysis
- isComponentPlan: System Engineering master plan
- isInputTo: Program technical development Plan
- hasOutputs: list of safety critical functions, hazard risk assessment, safety risk assessment
- Requires-Inputs: structural design analysis, FMECA
- Responsibility: office of chief engineer

Figure 2: A Plan template.

Figure 2 is a template that was derived from an analysis of example plans for multiple aircraft programs produced by a large aerospace company. The diagram illustrates a schematic display of information about a safety plan. The form has fields which have been filled in for the System Safety Plan. Analysis of the kinds of entities referenced and the relationships between them can help build specifications for these artifacts which can make program evaluation much more quantitative and grouped in reality. These specifications will be part of the ontology.

A good work product specification identifies components of the work product and relationships to other work products. For example, a plan may call out subplans, other products entailed such as reports. System specifications need sufficient detail so potential consumers can tell whether this work product is the source of the information that they need. Data is highly interrelated as the production of almost all data requires other data as input. Specifications for the data content require the ability to identify relationships and properties of a physical object that can be measured, analyzed, and tested. For example, the work products documenting the results of an air frame external structural loads analysis may need to specify the analysis methods to be used for a specific milestone.

By looking at the totality of digital documents for a program one recognizes that all of the documents contain common metadata regarding author, data of production, revision, etc. Most information system keeps this kind of data for retrieval and to document the provenance of an individual copy of the document. By examining a kind of document such as plan one finds considerably more commonality regardless of the specific kind of plan such as a safety plan. As one would expect the more specialized the kind of plan the more commonality one finds. The System Safety plan applies to all end products and applies to the full lifecycle of each product; plan execution has objective and outputs. The list of safety critical functions is a task output of System Safety. Many of the fields are common to other plans.

In Figure 2 the field values are in some cases other digital products (e.g., the system engineering master plan), some cases classes of product components (e.g., all product variants), tasks (e.g., hazard analysis), activities (e.g., structural design analysis), individual roles (e.g., chief engineer), and methods (e.g., fault tree analysis). For program management one needs program level agreement on what these fields and their admissible values are for at the major work products. One needs to be sure that this information is present in the safety plan. From one work product that references another work product the user needs to be able to follow the links to additional referenced work products. This is where a product development ontology can help.

## 3 CONCEPTUAL MODELING

Plans are a good place to start for constructing a product development ontology. The plan template in Figure 2 uses a terminology of physical equipment, software, aircraft components, test situations, outcome events, methods, as well as other plans and documents. From the conceptual modeling perspective, or a software perspective, the fields in the safety plan are attributes whose values are defined for classes.

Classes are simply a way of classifying different

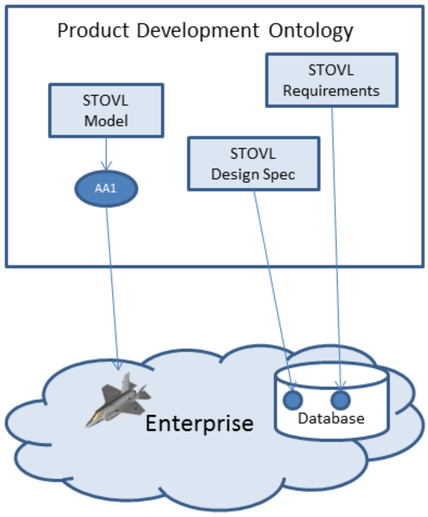


Figure 3: Relationship between conceptual model and subject.

kinds of entities. The classes of the ontology will classify both physical objects such as a specific aircraft and information objects such as requirements and design specifications. A specific plan is an instance of a plan class. These classes and their relationships constitute a conceptual model of aircraft development environment. Each class categorizes the kind of objects being talked about. Some of the objects are physical objects and some are digital objects. An ontology for product development is a classification of concepts and relations between physical and information objects. The ontology classifies physical objects and events, as well as, work products (information objects) such as plans and the work products called out by the plans.

Figure 3 illustrates the physical enterprise which is the subject of the ontology and the model of the ontology. The ontology describes both physical artifacts, classes of physical artifacts, as well as digital artifacts. All of these artifacts exist or potentially exist in the enterprise. The diagram contains three classes, $STOVLModel$, $STOVLRequirements$, and $STOVLDesignSpecification$. This model also has a small circle labelled $AA1$ which stands for a physical aircraft. The model object $AA1$ stands for the physical aircraft illustrated in the cloud representing the enterprise. The enterprise also has a database indicated by the “oil drum”. The database contains digital artifacts which correspond to the instances of the three classes in the ontology. From the diagram one can not tell whether any requirement or design specifications exist. The diagram shows the $AA1$ aircraft as existing. An interesting question that we will look at is when can one add $AA1$ to the conceptual model.

The Plan class most likely a specialization of a more general document class, and the plan class may have specializations. The value types of the class attributes such as Plan, Hazard Analysis, Fault tree analysis are themselves classes. These classes and the relations between them constitute a product development ontology. These classes or concepts in conceptual and roles in modeling terminology are common to many domains. The next section indicates how these concepts have been well worked out and can be adapted from a foundation ontology. Use of an established foundation ontology can both save considerable work and can help achieve standardization across a product development domain. The plan classes can serve as templates or specifications for plans if they have sufficient detail.

The product development ontology models or describes the product development enterprise. For example it contains classes for physical objects, their digital requirements and design specifications, as well as analysis and test results for specific aircraft. A success criteria for a product development ontology to be used effectively by stakeholders interacting with an information system organized by the ontology is that from the classes used to classify aircraft of a particular model type, one can follow links to requirements and specifications for that model type, and one can connect both to the physical reality of the enterprise. This means that the product development ontology constructions need to make the mapping of the ontology to physical reality sufficiently precise and simple that there are no ambiguities regarding, for example, whether a physical aircraft is a member of a specific class.

For the ontology to be effective for program management the describes relationship in the Figure 3 diagram must be sufficiently simple and direct that stakeholders can correspond the identification in a digital artifact with the physical subject. To create an ontology that can be used by an information system the model has to be represented in a formal modeling language that can produce output that can be used to configure an information system.

From the ontology perspective a specific air vehicle with a unique identification number is a member of a class of aircraft. The aircraft may also may be a member of a class of aircraft which have been validated as meeting the specification for a specific model variant. Classes can be introduced for model variants. All of the aircraft model variants could be represented as an instance of a class for a specific kind of variant. A specific safety plan for an aircraft program is a member of a safety plan class as is the design specification for the aircraft model.

The physical objects are the product components, the equipment artifacts used to produce and test the product, and the operational environment for the product. Material, events, personnel, tasks and processes are all part of the physical reality. The descriptions include the physical processes, tasks, and the results of performing the tasks. To compare product development activity with resulting work products an ontology represents physical reality including products and their components, and events. The digital work products include specifications and descriptions for physical products and results of analysis and measurement related to the physical products. For a product development ontology to provide stakeholders with results that allow effective assessment of development progress to be measured the ontology has to be integrated with the program information management system so that a user can explore the ontology to understand what digital work products are associated with which digital and physical entities.

#### 3.0.1 Foundation Ontologies

Many of the general concepts needed for product development, such as task, plan, product, and event, are domain independent and readily exist in foundation (upper level, core) ontologies, such as DOLCE (Gangemi et al., 2002) and ABC Ontology (Lagoze and Hunter, 2001). A foundation ontology characterizes general terms, e.g., entity, event, spatial and temporal location, and basic relations, e.g., part-of, quality-of, participation, and dependence. These concepts are well worked out, in use, and provide a good starting point for product development ontology. A foundation ontology with extensions for the product development domain can, once it is developed, be used across multiple product development applications. Its classes may require additional specialization for specific applications.

The diagram in Figure 4 is a Protege display of a DOLCE class inheritance hierarchy. Some of the general classes are: $PhysicalObject$ which is used to classify objects that occupy space and have mass and $Quality$ which is used to classify entities that are perceived or measured of a physical object. Examples of qualities are color, length, mass, and shape. These classes are used to classify the actual values which measurements result in. Event is an inclusive class for entities that occur in time. The first flight of the first aircraft of a design model is an event. While an event is an entity, it is not an object. Process is a subclass of event that describes transformation of inputs to out-

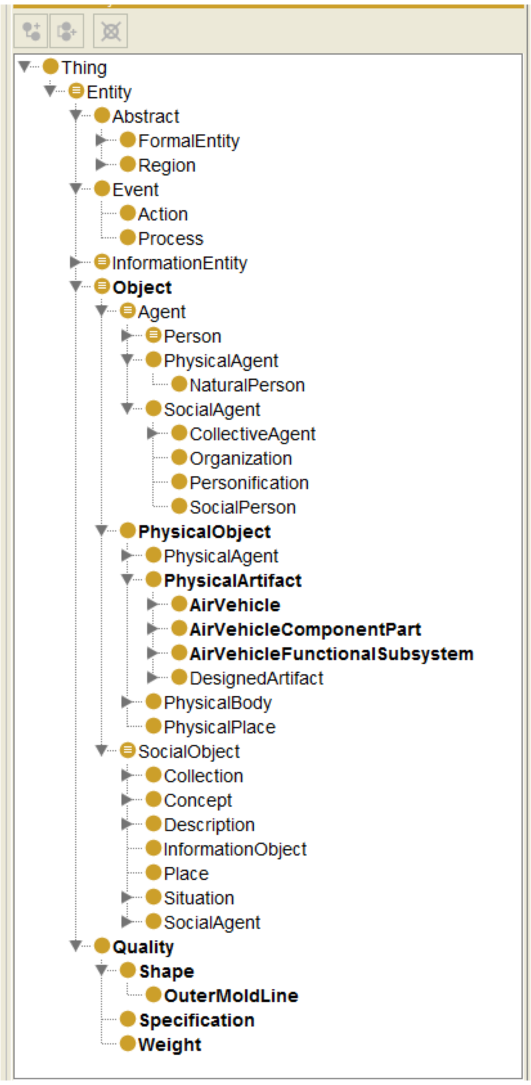


Figure 4: Part of the DOLCE class hierarchy

puts. Events can be described in different ways, possibly based on achievements, or on typical participants (e.g. human and organizational agents and physical objects).

The Description class is the disjoint union of classes that are useful in classifying work products. *Description* includes as subclasses, Contract, Design, Diagnosis, Goal, Method, Norm, Plan, Relation, and Right. For example, a plan is a description of some actions to be executed by agents in a certain way, with certain parameters; a Diagnosis is a description that provides an interpretation for a set of observed entities, etc. A method is a Description that defines or uses concepts in order to guide carrying out actions aimed at a solution with respect to a problem. It is different from a Plan, because plans could be carried out in order to follow a method, but a method can be followed by executing alternative plans. A digital plan is an $InformationObject$ as well as a Plan.

Classes are connected by relations. For example a description is related to the physical world through the satisfaction relation. The satisfies relation connects physical object with descriptions. A situation can be conceptualized as an accomplishment (which has brought a certain state to occur), as an achievement (the state resulting from a previous accomplishment), as a punctual event, or as a transition (something that has changed our focus from a state to another). Shifts from one aspect to another include: a) causation focus, b) effectual focus, c) transition (causality).

System engineering practice involves assigning quantifiable values to physical object properties. Physical properties connect mathematics to the physical world. Statements about physical object properties may express product requirements, such as a weight restriction or average fuel consumption, or may be the result of observing and measuring property values, or be derived from observations. DOLCE includes as class Quality and properties which relate physical objects to quality instances. A quality class is coordinatized by a mathematical space which structures the possible values that an observation of the property may yield.

## 3.1 How Foundation classes are used

Figure 5 is a graphical representation of part of a product development ontology. It illustrates relations between classes of physical objects, qualities, and the situations in which measurement results are obtained. The diagram illustrates the interrelationships between the products which are designed, analysed, measured, and tested, are results of tasks, the properties of products being developed, and the results that are obtained. The light blue rectangles are foundation ontology classes. The blue rectangles contained in the gray ones are product domain classes. The blue boxes are classes which have instances. The ontology may have instances such as $AA1$.

The AirVehicle rectangle represents a product class rather than an individual aircraft. The class has a specification or description which characterizes instances of the class. The model class is a conceptualization of physical reality. The conceptualization in this diagram includes component classes such as air frame, avionics, and systems. Similarly the design, manufacturing, and test equipment are conceptualizations. These conceptualizations are used by stakeholders in the development process to communicate about the physical world. For these conceptualizations to be practically useful in connecting the

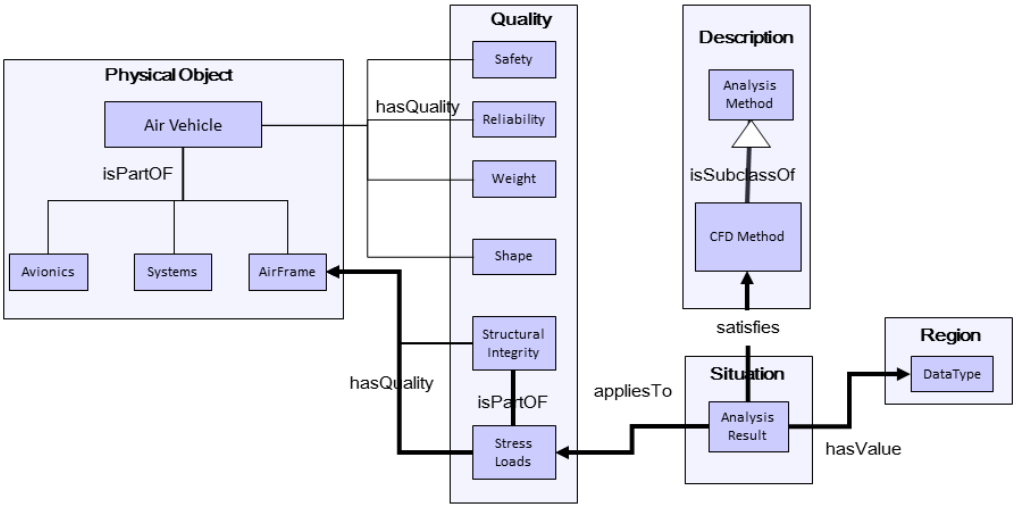


Figure 5: A part of the conceptual model.

conceptual world to the physical world requires constructing conceptualizations for which the stakeholders have the means to determine whether a real world satisfies the conception. For example, the question of whether a specific physical aircraft is a member of the model class must be determined by a customer when he takes delivery. In some cases when delivery of an aircraft is made there is an inspection made to ensure that it has the specified equipment.

A class such as *AirVehicle* in Figure 5 is used to represent the physical air vehicles which satisfy a particular specification. The class would likely be part of the development program's ontology from the beginning. The membership criteria for the class would reflect industry practice. For example, membership criteria for this class likely requires that a detailed design specification has been signed off on, that a vehicle identification number has been assigned, and that the task of building the vehicle has been planned and scheduled. The membership criteria do not necessarily imply that the *AA*1 aircraft has actually been built.

An air vehicle (model) is classified as a physical object as are its physical characteristics. However, a description of the air vehicle such as requirements and design specifications is an instance of the description class intersected with the information artifact class as are descriptions of the processes for measuring air vehicle characteristics. The characteristics of an air vehicle such as Safety, Reliability, Weight, Signature, and Structural Integrity are subclasses of the class, Quality. Stress Loads and Structural Integrity are both subclasses of the class Quality. Stress loads are part properties of Structural Integrity. These classes have properties whose values are methods of calculating values for the quality.

In Figure 5 the arrow connecting *AirFrame* to *StressLoads* represents the *hasQuality* relation between the two classes. Each air frame object may be related by the *hasQuality* relation to multiple stress loads objects. Each stress loads object is related to a single description of how it was obtained, and to the values that were obtained. To determine if *AA*1 is ready for first flight these links will need to be explored in more depth.

## 3.2 Analysis Results, Analytic Setups, Methods

An engineer assessing the readiness for first flight may want to determine what results for stress loads for the airframe object have been obtained, as well as how and when they were obtained. Following the relationship from stress load analysis results to the description of the analysis set-up may be sufficient. However,one may want to explore additional information that is known about the physical analysis process

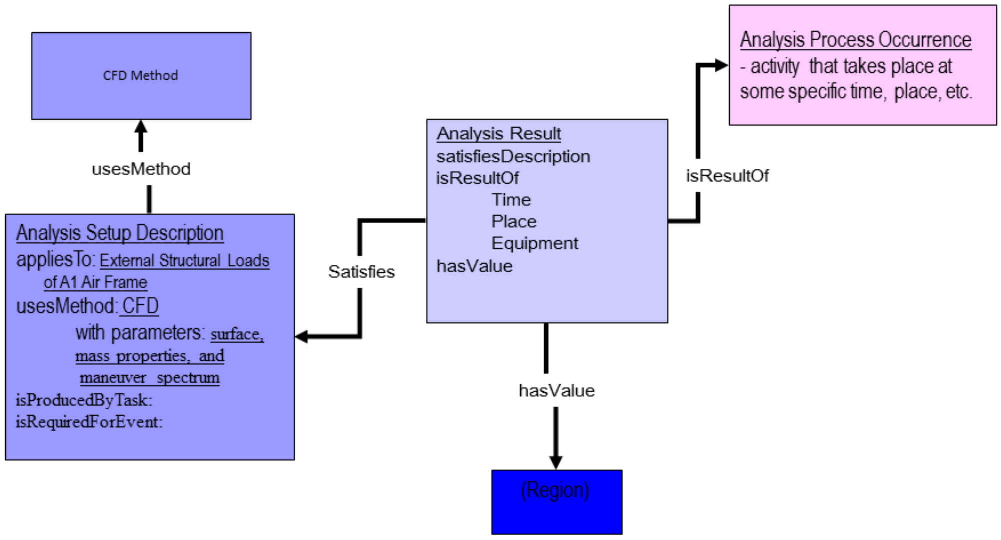


Figure 6: An analysis reuslt.

occurrence as documented in its digital artifact.

The diagram in Figure 6 distinguishes between the analysis setup description, the analysis result which provides parameter values for data obtained, and the analysis process which takes place in time and produces the data. The diagram in Figure 6 illustrates a work product specification for an analysis result in the form of a description of the analysis setup. The description identifies that the analysis is about the structural loads of the AA-1 airframe and uses a computational fluid dynamics method. The analysis set up is linked by the satisfies relation to analysis results which satisfy the analysis setup description. There may be multiple work products that satisfy the description.

An analysis result is a view of some analysis process occurrence and is described by the analysis setup. Detailed planning for a task, such as product verification, will produce descriptions of the analysis to be performed, e.g., of the setup and methods which then get executed and the results recorded. The descriptions are used to evaluate the situations that provide a view of an actual event. The results of an analysis of the external stress loads classify an analysis process execution. The process occurrence is an event that occurs in a time interval. An analysis is a process, a controlled study to discover, evaluate and/or verify safety, effectiveness, and compliance of some quality. The analysis process may be composed of numerous sub-processes, carried out by agents planning specialized roles for the production of services needed for some conclusion. The analysis result satisfies, or is described by, a description of the process used to produce the analysis result. The analysis process description is about calculating the external structural loads of an air frame. The specific analysis result is about the Structural Integrity Quality of the product *AA*1.

## 3.3 Status Assessment

Execution of the development process and its plans and work flows entails the creation of instances of classes. For example, a program system safety plan is an instance of the plan class. The plan will require creation of specified work products. A conceptual model imposes a class structure on the perceived entities in the world. Each class has properties defined for the members of the class. Classes and Relationships define specifications for the members of the class.

Before the first flight of the initial air vehicle (an event) a number of preconditions have to be met. These conditions include ascertaining that specific analyses have been performed and the results of these analyses meet specified criteria. A plan for first flight will identify airframe, avionics, and subsystems such as hydraulic and electrical have to have what test results. The subsystems are represented in the model of the aircraft. For subsystem and each component will have identified measurable qualities such as shape and weight, for which there are expected to be measured results. In Figure 5 Structural Integrity is a class which has as a part component Stress Loads. If a consumer wants to identify what tests have been performed and what were the results, this can be done by following the links from Stress Loads to the Stress Loads Analysis Results. By identifying the results one can trace to the description of the method used and the results obtained.

A stakeholder by viewing the part of the ontology relevant to a task such as assessing readiness for first flight can get as much detail as is described by the ontology into the state of the development process. By drilling down the stakeholder may of course need to check what has happened in the physical enterprise and see if it appears to correspond to the ontology information picture.

# 4 SUMMARY AND FUTURE DIRECTIONS

The argument made here is that a product development ontology offers a path to enable a development program to quantifiably measure how well the program is succeeding in meeting its objectives. While not much as been said about measuring success one can see that one can measure the number of accepted work products against what has been specified in the program plans. Further at level of management one can have some level of assurance what reality the results correspond to.

Effective program planning and execution depend on having good specifications for work products. A conceptual model (ontology) for product development concepts and interrelationships can be used to produce work product specifications. Well worked out foundational ontologies can be extended to produce product development conceptual models. A product development information system can be designed to be configured with a conceptual model and provide services not available in traditional data management systems. Considerable work is required to construct a product development ontology with full coverage. In particular the construction of detailed work flows and work product specifications is labor intensive. However, such work is necessary the execution of product development processes to be properly assessed. An enterprise does typically spend effort to develop work flows and work product descriptions. However, without the formalization that can be achieved with a formal ontology the questions of whether the processes are being followed and are effective are difficult to answer.

A conceptual model of product development that classifies the different kinds of entities in product development and defines relationships between products, tasks, and work products can be used to establish work product requirements specifications. The entities that need to be classified include the actual products being developed together with their properties, as well as the work products that result from activities to design, analyze, measure, and test, as well as the plans of the tasks, workflows. The ontology formalizes the work of planning and can be used to minimize what users enter when work products are created. A conceptual model class definition can be used as a template for defining a work product.

The ability to implement information system that can be configured by an ontology is well within the bounds of current information technology. A product development ontology can be integrated with existing technical data management. product development. An ontology can be exported from a development tool such as Protg in an XML format. The XML format can be stored in an information repository and can then be used to dynamically generate web-forms. The forms can provide discovery, access, and query answering across distributed data repositories that support Web-services.

Some of the classes represent physical objects and events, and some classes represent work products (information objects) such as plans and the work products called out by the plans. The classes and relations of the ontology are common to the product domain, not just a single product. The ontology can be reused across programs and so will likely be done prior to program execution. Application to a specific program may require tailoring. The specific instances of the classes and facts about them are the information created during the execution of the product development program.

Good approximate representations of product development ontologies can be made in UML or SysML. These formal representations can then be used to generate data in a form that can be used to configure an information system. However,a more precise analysis of ontology language expressiveness and formal semantics based on the use cases provided by product development. With a firm formal semantics in place for the ontology automated reasoning based on the ontology has great potential for analysing the state of a product development effort (Graves and Bijan, 2012). Automated reasoning can check that specifications exist for each work product, check consistency of tasks, check that descriptions of work products are consistent, and check that test results cover test conditions.

## REFERENCES


Borst, W. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening ontologies with dolce. *Knowledge engineering and knowledge management: Ontologies and the semantic Web*, pages 223–233.

Graves, H. (2007). Ontology engineering for product development. *Proceedings of the Third OWL Experiences and Directions Workshop*.

Graves, H. and Bijan, Y. (2012). Using formal methods with sysml in aerospace design and engineering. *Annals of Mathematics and Artificial Intelligence*, pages 1–50.

Graves, H., ODonnell, C., and Campbell, R. (2004). Using the us department of defense architecture framework (dodaf) in reengineering product development information. *SETE 2004: Focussing on Project Success; Conference Proceedings; 8-10 November 2004*, page 247.

Kim, K., Manley, D., and Yang, H. (2006). Ontology-based assembly design and information sharing for collaborative product development. *Computer-Aided Design*, 38(12):1233–1250.

Lagoze, C. and Hunter, J. (2001). The abc ontology and model. In *DC-2001: International Conference on Dublin Core and Metadata Applications 2001*, volume 2, pages 1–18. British Computer Society and Oxford University Press.

Scrudder, R., Graves, W., Teigen, T., Johnson, C., Hix, S., and Hollenbach, J. (2003). The critical role of metadata in jsf development. *2003 Fall Simulation Interoperability Workshop*.

Sudarsan, R., Fenves, S., Sriram, R., and Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer-Aided Design*, 37(13):1399–1411.