

Automating Process with Knowledge Templates

Ken Versprille, PLM Research Director • February 17, 2005

In the product development world, competitive advantage drives profitability. Harnessing a company's intellectual property and product know-how and then targeting it to improve efficiency and productivity builds competitive advantage. The challenge is to identify, capture, and institutionalize optimal process flows that promote greater concurrency between product design and other domains within the corporation, including manufacturing, support, finance, procurement, sales, and marketing. The emerging technology of knowledge templates focuses on delivering the needed tools into the hands of product developers to capture knowledge, and to formalize their use.

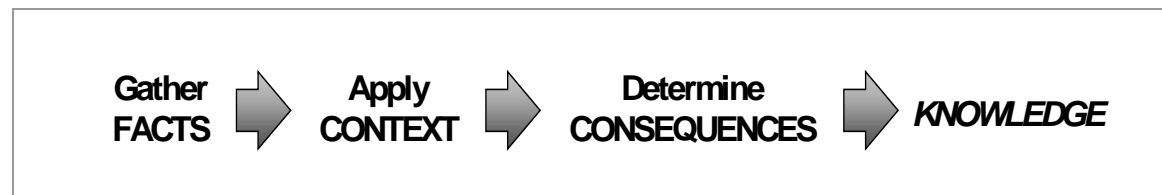
Key questions must be answered. What are knowledge templates, and which end-user processes are best suited for automation through their use? What pitfalls in process selection should be avoided? Further, what capabilities should be considered in making the proper choice of a knowledge process tool? And finally, how does a company establish a methodology for automating process and then promote its use within the organization?

Knowledge-Based Engineering

Knowledge templates live within the broader context of what the industry has come to call "KBE," or Knowledge-Based Engineering. Many question, however, whether KBE is the best terminology, choosing instead to use the phrase, "knowledge process." In many ways they are correct, since the goal of the technology is to intelligently capture process.

Knowledge-Based Engineering can be loosely defined as the use of varied software tools and techniques that capture product and process knowledge. Capturing knowledge first requires that facts be gathered about each product step being automated by KBE techniques (Figure 1). Facts can be specific numbers such as lengths, widths, or weights, or a range of numbers. For example, an automobile gasoline tank may require a volume between twenty and twenty-two gallons, or the total material cost of a product must be less than a specific dollar amount. Facts can also be attribute information such as the material type of the product or any one of its components.

Figure 1: Steps to Capturing Knowledge



CPDA: Collaborative Product Development Associates, LLC

CPDA's Design Creation and Validation program provides this Technology Trends in PLM to all of our collaborative research sponsors. Those interested in the program should contact cust_service@cpd-associates.com. CPDA thanks IBM for funding this research and report.

Automating Process with Knowledge Templates

February 17, 2005

Next, a context must be applied that relates the data items and determines the consequences of that context. For example, when dealing with a particular design object composed of steel that serves both a functional and a structural purpose, that context might imply a consequence that the object must be of a particular minimum thickness.

To transition from facts to knowledge, comparisons must be made between different sets of information and prior experience. That experience is then broadened through in-depth interviews with subject matter experts, such as “domain experts” in procurement, manufacturing, or simulation. Only then can connections be made and consequences be determined. In essence, the process takes raw information and provides another level of context. This process of identifying relationships and drawing conclusions is the creation of knowledge.

Once all this is done, that implicit knowledge must be transformed into explicit knowledge by codifying the steps. Methods of codifying knowledge depend very much on the software technology employed. They can range from classic programming at one end of the spectrum, all the way to simply sketching graphic flowcharts, depending on the tool chosen. A specific tool will likely bound the flexibility of a knowledge application.

The purpose of creating knowledge-based applications is to ensure and formalize their repeated use in the product development process. An important point in the definition of KBE is the phrase, “in an integrated way.” Study of leading-edge users of knowledge engineering technology has shown that the largest impact has been felt when users integrate knowledge between two or more diverse applications. Recent events have showcased examples, such as an automotive power train user integrating design with CAE analysis, to help drive design variation based on simulation results. In another example, a product development organization integrates design, finance, and procurement to better select off-the-shelf components, reducing product cost. And in the aerospace industry, a company integrates the design and manufacturing process for mounting structures for the production assembly of composite product components. Each knowledge project has not only resulted in dramatic time and cost savings from these automated applications, but has also benefited from a reduction of errors in downstream manufacturing processes and other domains previously ascribed to a lack of communication.

The Components of KBE

Digging deeper, three components within KBE can be isolated. The first being the *objects*. These are the facts that must be uncovered about the product and the process. The second is *rules*. Rules are the discreet operations offered by the applications being automated. In a CAD application, a rule could be the creation of a fillet or the placement of a hole form feature. Alternatively, in a CAE application, a rule might be a meshing operation. Each application rule takes data values or objects as input and produces an outcome.

When discussing KBE rules, many often make a distinction between *process-centric rules* and simple CAD *parametric rules*. For example, if the concern centers on placing a properly sized hole form-feature, computing its diameter based on a specific formula is a CAD parametric rule. However, if material type must also be considered to determine whether the hole can be punched or must be drilled, based on the combination of hole diameter and material properties, users cross into process-centric rules.

The third component of KBE is *knowledge process modeling*. Knowledge process modeling is the act of using product and process knowledge to put the rules of the application or applications together and properly sequence them so that they consume data objects and evolve them step-by-step as the rules are applied. This is corporate competitive know-how and product engineering expertise.

Rules have *active* intelligence. When sequencing through an application toward an outcome, decision points are reached: “Do I go left or do I go right?” Depending upon the particular application, that rule could be very specific to the product or process under development. Process modeling on the other hand, refers to a company’s business process or business framework. Not every company will use the same vocabulary, but within process modeling there are *passive* rules – meaning their intelligence never changes the way rules change at the active level.

Knowledge Templates

The secret to success in Knowledge Engineering is “manageable consistency.” The one cry for help constantly heard from management and users alike across all the industry verticals, from automotive, aerospace, large machinery, on into consumer products, is the need for consistency. Each argues that there is a best, proper way to accomplish a specific task given their experience and type of product. Yet each is often handed inconsistent data, or worse missing data, for performing that task. This leads to their inability to confidently select the proper design and manufacturing procedure, leading to potential errors and delays downstream in the process.

Any organization can decide upon and document proper procedures, but how sure are they that those procedures are actually being followed? How do they communicate any new agreed-upon improvement to a process? Where do they communicate it? Management is looking for repeatability to minimize risk, looking for assurances that approved processes have been followed. Users spend countless hours trying to confirm that the initial data that they have been handed is correct, complete, and has passed company standards. And consider the impact to a program or project manager if, when a correct procedure has been followed, it automatically records that fact in their project management application.

The technology of knowledge templates (or knowledge scripts) offers a solution to the requirements of “manageable consistency.” Templates meet a global requirement of

Automating Process with Knowledge Templates

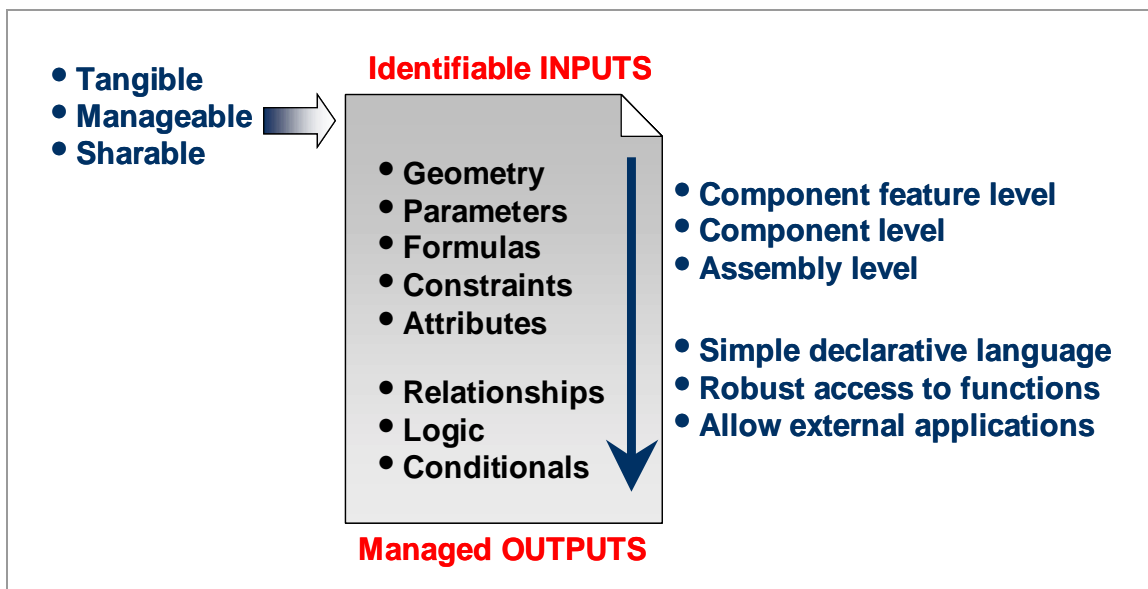
February 17, 2005

capturing a knowledge-driven process in a single-source, structured location whether encoded as a file or other similar structure. It offers a single, consistent location within which an improved process can be articulated. It exists in one place. It can be versioned and data managed. It can be archived in a central repository and shared across organizations.

Knowledge template tools, however, vary. Based upon research into the use of knowledge templates, as well as the need to fully meet diverse end user requirements, users are advised to look for a detailed list of capabilities and qualities in their choice of template tool.

First, the template tool must offer clean identifiable inputs for its use (Figure 2). Its execution must also result in “*managed* output results.” That is, results that remember they were generated through the use of a given template. That way, if a product requirement changes, it is possible for users to trace through which arguments and which templates have been used and need to be rerun. It is necessary to recognize, however, the part model file now has a dependency on the template file.

Figure 2: Knowledge Template Capabilities



Internal to a template, authors of the automated process must be given flexible access to a full range of application data. In CAD, that data could include geometry perimeters and formulas used to generate the geometry, but also geometric constraints and object attributes. Additional access and controls must exist to query and establish new relationships between data items. And a robust set of tools should be available for process flow logic and conditionals.

Data access for read and write is best served by using the application's formal Application Programming Interface (API) that hopefully provides some buffer between changes made to the application release by release. Further, a knowledge template should be able to operate on various product design structures within a CAD model (or other application) such as at an individual component model feature level, a full component model, or an assembly of models.

The programming language authors use to code their processes should be straightforward and easy to use. Any proprietary language should be patterned as close as possible to a *de facto* standard such as Visual Basic for broader appeal. Many companies will make a decision to employ engineers as knowledge application authors, rather than software programmers. Engineers can converse more readily with subject matter experts, but they require an easy-to-learn and -use programming tool. As their skills mature, it may be possible to escalate to a C or C++ programming language.

Finally, for a fully integrated capability, the template language should allow integrated access across diverse applications, such as CAD or CAE, and allow triggers to execute external applications supplied by other vendors.

Choosing Knowledge Applications

Experience with leading-edge end users has shown that KBE tools apply best to well-defined processes. Inexperienced users should embark by first selecting their most redundant activities. These are typically the activities that the experienced engineer hates to do because they are repetitive, procedural, and mundane. Avoid automating processes that are not well established or that require a human judgment. For example, automotive styling would not be considered a good target application. Too much subjective input goes into what makes a well-styled vehicle.

One important consideration in the deployment of any knowledge application is user buy-in. This opens a large issue of impact on culture. Building a number of user decision steps into an automated knowledge application often helps with user acceptance.

In developing an automated process, grow skills in building and documenting what in effect is the logic tree of a process flow. Each step in the process must have a clearly articulated understanding of what data is accessed, how it is manipulated, and all possible outcomes. Then flowchart the logic necessary to control and manage through the sequence of each process step. Be sure to record the *why* of each particular step in the process. Such information will greatly assist in any future changes made to the process.

Automating Process with Knowledge Templates

February 17, 2005

Summary and Recommendations

Companies that actively pursue knowledge engineering tools today must first expect that those they designate as the authors of knowledge applications will face challenges in gaining the needed skills. They must learn how to quickly gain the trust of subject-matter experts and they must become skilled at extracting knowledge from those experts, not just facts. They must assimilate only relevant knowledge, often making choices when experts disagree. And they must learn to codify that knowledge using the tools provided by the knowledge engineering solution the company deploys by converting the knowledge from a form understood by humans into a form understood by the computer.

Automate only well understood processes. Redesigning the product or its process is best left to the subject-matter experts.

Asking the right questions is a skill developed over time. As experience is gained, create a logic tree of the application while interviewing subject matter experts. Following that logic may uncover holes in the information the experts are providing or potentially identify missing steps.

Continuously ask questions. Subject matter experts will often leave something out simply because to them it is unimportant or they do not understand what is being asked of them. Maintain an ongoing communication with experts during the application authoring process in order to continuously capture hidden knowledge.

Authors must also understand the entire process. They need to identify all the requirements. Missing knowledge will obviously impact the outcome of a knowledge application. And finally, authors of automating process should understand that subject matter experts will be as weary of them as they are of each other. Authors will often find themselves in a unique situation as they try to automate a process. Procedures will be questioned and wasted time and effort will become very apparent. Many in the organization will want to cling to a regular routine. An automated knowledge engineering process implemented through templates needs the facts. The job of a knowledge template author is to obtain the facts, get the experts to agree, and work collectively towards the best process.

This document is copyrighted © by Collaborative Product Development Associates, LLC (CPDA) and is protected by U.S. and international copyright laws and conventions. This document may not be copied, reproduced, stored in a retrieval system, transmitted in any form, posted on a public or private website or bulletin board, or sublicensed to a third party without the written consent of CPDA. No copyright may be obscured or removed from the paper. Collaborative Product Development Associates and CPDA are trademarks of Collaborative Product Development Associates, LLC. All trademarks and registered marks of products and companies referred to in this paper are protected.

This document was developed on the basis of information and sources believed to be reliable. This document is to be used "as is." CPDA makes no guarantees or representations regarding, and shall have no liability for the accuracy of, data, subject matter, quality, or timeliness of the content.