The Role of Deliverable Specification In Automated Process Planning

Debra Walker
CEO - Apprentice Systems Inc.
6189 Wessington Dr.
Hudson, OH 44236
dwalker@apprenticesystems.com

The Challenges of Distributed Planning and Scheduling

When the challenge is to design the most effective and efficient process for producing a deliverable (whether it is a physical product, a system, a document, or a bundle of services), the complexity of the deliverable itself influences the complexity of the process. Complex products require complex processes, involving multiple agents with different skill sets and domain knowledge. Moreover, complex products that are optimal solutions seldom result from processes that merely adopt a "least commitment" approach – that merely eliminate possible solutions on the basis of constraints. Producing optimal deliverables from efficient processes requires a coordinated approach to deliverable planning. A useful deliverable specification can provide a shared input to distributed rule-based planning tools, which can then coordinate the efforts of multiple human agents.

A Rule-Based, Generative Approach to Deliverable Planning

A detailed specification of the deliverable is a critical input to planning and executing an efficient process, but that specification must be easily adapted, must be feasible, and must be shared. Generating a detailed specification of a complex product using traditional methods is neither easy nor quick, despite the demands of real-world timelines. Detail is essential, however, if the specification is to be useful. Moreover, an initial specification that is complete is undesirable if planners cannot easily adapt the product design during the production process. And unfeasible high-level goals inevitably lead to midstream modifications in sub-goals, tasks and resources, making coordination difficult if not impossible. The process must vet the specified output for feasibility – both at the outset and throughout the production process. Finally, a specification that truly informs the process must be shared (as appropriate) by all agents participating in the process.

What is needed to provide a detailed, adaptable, feasible, and shared deliverable specification in a timely fashion is an automated collaborative design framework. This automated framework must make it easy for all the stakeholders of the process, including the agents who must execute it, to incorporate their knowledge into the design. This knowledge is not just traditional product configuration knowledge (e.g. which technical solutions meet which use requirements) but also process knowledge. A familiar example of a complex product is tires. Product knowledge blends seamlessly into process knowledge: certain types of tire tread designs (e.g. asymmetric) require certain types of molds (e.g. segmented), which require certain types of models (e.g. laser-engraved). This knowledge indicates not only what type of tire is feasible given equipment availability, but also what types of information are necessary at various points in the process (bills of materials, drawings, data) and may need to be generated as part of the product specification.

The interdependence of product and process (and therefore the desirability of collaboration between designers and implementers) has been understood for some time in the manufacturing industry. In the 80's and 90's this collaborative approach was dubbed 'simultaneous engineering' and pursued as a significant improvement to product development in the automotive industry, but with little success because the attempts used traditional design and project management tools.

In recent years, however, technologies have been developed that truly enable automated collaborative design, and successful tools have been built in manufacturing, retail, service and non-profit industries.

This proven approach to collaborative design automation utilizes diagramming tools and wizards to capture both product and process knowledge in a Knowledge Model. The model is then used by an event-driven inference engine to generate an automated design tool that incorporates that knowledge. This inference engine has been optimized for reasoning over semantic networks.

The core of the Knowledge Model contains traditional 'expert system' trees – composition trees containing roles connected by part/subpart relationships, and content trees containing concepts connected by class/subclass relationships. Important extensions beyond these basic networks include mechanisms for describing and monitoring the model context and mechanisms for managing the interface with the user (both inputs and outputs). Event-driven rules link all these components together.

The rules have been optimized for reasoning over semantic networks and are tailored to monitor and drive the six basic types of transactions that happen in such a network: the creation and elimination of nodes, the connecting and disconnecting of nodes, and the setting and reading of attributes. The timing of event-driven rules in the system is driven by triggers that 'listen' for the same types of changes: nodes being added or eliminated, relationships being created or destroyed, and attribute values changing.

Knowledge Models with this structure are self-contained yet modular – one model can include and extend the ontology of another model. This modularity is especially useful when dealing with a complex product or process that requires inputs from many different groups of people. The various Knowledge Models can be built and maintained separately but combined at runtime to create a high-level, integrated model.

This collaborative design framework meets the requirements laid out for deliverable planning: detailed specifications can be automatically generated based on the current context; the format of the output (reports, data, drawings) can be specified in the knowledge model and custom-tailored to agent requirements; the resulting design is feasible based on the knowledge of all the participants, including the process agents; the design can be regenerated by changing context values as circumstances change; and both the outputs and the responsibility for the knowledge can be shared through modular Knowledge Models.

Broad Application of the Meta-model

Because this approach utilizes a meta-model capable of reasoning over any domain that can be described via semantic networks, it has great flexibility and has been used to model complex products, systems and services as well as documents, legal arguments and other deliverables not typically viewed as 'products'. This flexibility makes it possible to generate a detailed specification for any process deliverable, and enables a teleological approach to processes that are not currently thought of as producing a tangible, and therefore specifiable, deliverable.

A Rule-Based, Generative Approach to Process Planning

Even with exceptional upfront planning, including a detailed and feasible specification of the final process deliverable, midcourse corrections will be required to respond to uncontrollable changes. The same collaborative framework used for deliverable design can be used for automated process planning. Process knowledge is captured and used to generate an event-driven, rule-based Hierarchical Task Network (HTN) model capable of generating custom processes in response to a set of context variables. A high-level goal, such as "introduce a new tire", is broken down into sub-goals (e.g. design the tire, build the tire, market the tire) which are then parceled out to sub-models for decomposition into a partially-ordered set of primitive tasks (a "least commitment" approach is used in this step).

The sub-models are built based on the knowledge of the agents responsible for that portion of the process. Rules based on conditional logic are written to determine which sub-goals and tasks should exist/not exist, which dependencies should exist/not exist, which and how many resources should be assigned, and how much work and time are required.

The conditions that drive the rules come from the Context of the model, and setting the values for the context variables triggers the automatic generation of a custom plan to address that unique set of circumstances. In this way the plan, for the whole process or for portions of the process, can be quickly regenerated to address changes in requirements or environmental conditions.

The Interdependence of Product Planning and Process Planning

The interdependence of the specification of the product and the plan necessary to deliver it becomes clear when either product or process models are built. Different organizations may begin the modeling process from different starting points, but experience shows that they converge on the same place – an integration of the two.

Actual projects to create process automation tools have proven that the majority of 'context variables' can be pulled directly from the deliverable specification. For example, context variables in a new tire introduction process model will include the number of tire sizes, type of molds, and the sophistication of the sidewall design – all of which are part of the deliverable specification. Many organizations then move on to create an automated product model that can generate these context values reliably and quickly based on the expertise of the entire team, rather than leaving them to the judgment of the individual planner who must fill them in when running the process automation model.

Organizations that build automated product design models immediately see the value of that model's outputs as inputs to downstream processes – in fact, major process improvement efforts have been initiated based on the opportunities created by the ability to generate a detailed and feasible specification upfront. Often, one of the primary justifications for investing in an automated design system is a reduction in the rework that is created by infeasible designs entering the system.

Videos of this approach, both product and process modeling, can be seen at www.apprenticesystems.com. Also available are case studies showing the application of this approach at Motorola, Booz Allen Hamilton, REI, Cooper Tire, and a Harvard Business School case which references the product (Motorola) and process (REI) applications as two of four case studies representing a new breed of I/T applications.