J Intell Manuf (2017) 28:455–472
DOI 10.1007/s10845-014-0995-3

Decision guidance methodology for sustainable manufacturing using process analytics formalism

Guodong Shao · Alexander Brodsky · Seung-Jun Shin · Duck Bong Kim

Received: 12 August 2014 / Accepted: 29 October 2014 / Published online: 18 November 2014

Abstract Sustainable manufacturing has significant impact on a company's business performance and competitiveness in today's world. A growing number of manufacturing industries are initiating efforts to address sustainability issues; however, to achieve a higher level of sustainability, manufacturers need methodologies for formally describing, analyzing, evaluating, and optimizing sustainability performance metrics for manufacturing processes and systems. Currently, such methodologies are missing. This paper introduces a systematic decision-guidance methodology that uses the sustainable process analytics formalism (SPAF) developed at the National Institute of Standards and Technology. The methodology provides step-by-step guidance for users to perform sustainability performance analysis using SPAF, which supports data querying, what-if analysis, and decision optimization for sustainability metrics. Users use data from production, energy management, and a life cycle assessment reference database for modeling and analysis. As an example, a case study of investment planning for energy management systems has been performed to demonstrate the use of the methodology.

Keywords Decision guidance · Process analytics · Sustainable manufacturing · Optimization · Energy consumption

G. Shao · S.-J. Shin (✉) · D. B. Kim
Systems Integration Division, Engineering Laboratory, National Institute of Standards and Technology, 100 Bureau Drive, MS 8260, Gaithersburg, MD 20899-8260, USA
e-mail: seungjun.shin@nist.gov

A. Brodsky
Department of Computer Science, George Mason University, 4400 University Drive, MS 4A5, Fairfax, VA 22030-4444, USA

Introduction

This paper introduces a decision guidance methodology that uses the sustainable process analytics formalism (SPAF) developed at the National Institute of Standards and Technology (NIST) (Brodsky et al. 2014). The methodology is designed to help manufacturers achieve their sustainable manufacturing (SM) goals in a systematic and quantitative manner.

To be successful in today's complex, rapidly changing, and highly competitive world, manufacturers must comply with sustainability regulations that help them start sustainable practices for their manufacturing operations and stay competitive. The United States Department of Commerce (DOC) identified sustainable manufacturing as one of its high-priority performance goals, defining SM as the "creation of manufactured products that use processes that minimize negative environmental impacts, conserve energy and natural resources, are safe for employees, communities, and consumers and are economically sound (DOC 2014)." This means that manufacturers need to balance the environmental, economic, and social factors. Manufacturers' wise decisions to support SM could not only lead to significant savings for the company but also have positive impact on the society and environment. This paper focuses mainly on supporting the environmental and economic aspects of SM.

Many companies are employing practices that make their operations and manufacturing processes more sustainable (Fujitsu 2011; Philips 2012; GM 2013; Rockwell Automation 2014). They pay particular attention to improve their energy efficiency, because energy management is an important factor in almost all the sustainability issues. According to the US Energy Information Administration's energy outlook report, the industrial sector uses more energy than any other end-use sector, consuming about one-half of the world's total energy (EIA 2014a, b). A manufacturing energy consumption survey (MECS) shows that both the manufacturing energy use and the energy intensity of manufacturing activity between 2003 and 2010 have been reduced because of manufacturers' efforts (MECS 2013). EIA's Monthly Energy Review shows that energy-sector CO_2 emissions have also been lowered between 2005 and 2012. For example, the energy-sector CO_2 emissions were 8.3 % lower in January 2012 than in January 2010, and 12.9 % lower than in January 2005 (EIA 2014b). For example, Lockheed Martin's Go Green program indicates that a new Energy Management System (EMS) in Camden reduced carbon emissions by 2,332 metric tons per year (Rachuri 2010). SAIC has used its EMSs to realize significant facility efficiencies for some auto manufacturers in the US and monitor and control nearly 20 million square feet of facility with millions of dollars in savings (SAIC 2011).

Even though a growing number of manufacturing industries are initiating efforts on sustainability issues, many of them are targeting lower hanging fruit. Currently, most of the manufacturers' projects are conducted on an ad hoc and piecemeal basis for individual situations. The methods are customized, not reusable, and not extensible. The effect of many complex interactions is often not taken into account (Smith and Ball 2012). A systematic, quantitative, and model-based approach is required to reach a higher level of sustainability. The model-based approach involves modeling, simulation, and optimization. Increasing the availability, use, and effectiveness of modeling, simulation, and optimization technologies has been identified as a key enabler for improving Sustainable Manufacturing (SMLC 2011); many researchers have made efforts for optimizing specific manufacturing processes (Katherasan et al. 2014; Last et al. 2009; Ridwan et al. 2012; Wang et al. 2012) and improving energy efficiency and green logistics for manufacturing operations (Yang et al. 2013; Naeem et al. 2013). However, using such technologies, especially for small and medium-sized enterprises (SMEs), presents major challenges. These challenges, mainly due to the complexity of modeling and optimization, have led to the underutilization of these valuable techniques, and in turn, affected the quality and effectiveness of decision-making. One of the challenges to use optimization techniques is that it requires formal simulation and/or optimization models' development, which needs significant modeling expertise and substantial effort. There is a great need for making the modeling process easier and enabling reuse of modeling efforts.

To address the challenges discussed above, NIST researchers have developed the SPAF that allows users to model underlying reality once and reuse it many times for different tasks including querying, what-if analysis, and decision optimization (Brodsky et al. 2014). However, a decision guidance methodology of using SPAF for sustainable manufacturing has not been proposed. The focus of this paper is to fill the gap and introduce such a methodology. More specifically, the contributions of this paper are twofold. First, we propose a decision-guidance methodology that provides step-by-step procedures for performance analysis and assists sustainability improvement decision making in sustainable manufacturing using SPAF. This methodology includes the steps (1) problem identification, (2) knowledge and data collection, (3) formal problem representation, (4) what-if analysis and decision optimization, and (5) recommendation for improvement. Second, to demonstrate the developed methodology and formalism, a manufacturing case study is performed for deriving optimal resource allocation and process parameters.

The rest of the paper is organized as follows. "Related work" section discusses related work on industry SM efforts and decision guidance methods. "Overview of sustainable process analytics formalism" section briefly reviews SPAF. "Decision guidance methodology" section introduces the systematic SM decision guidance methodology. "Case study" section discusses a manufacturing case study to demonstrate the methodology. "Conclusion" section provides a conclusion and discusses the future work.

Related work

This section reviews decision support methodologies such as optimization and simulation as well as their applications and research efforts for SM.

Decision optimization and simulation Decision optimization has been used to find the best solution by exploring all the trade-offs. An Operations Research (OR) optimization model typically has (1) decision variables, (2) constraints that have to be satisfied, and (3) an objective function to be optimized. A feasible solution is an instantiation of values from corresponding domains of decision variables that satisfy all the constraints. Among all feasible solutions, an optimal solution is one that makes the objective minimal or maximal, as required (Brodsky et al. 2011). Using decision optimization to model a real world manufacturing problem is challenging because the model abstractions only have indirect connections to the elements of the problem. For example, one equation may include multiple parameters that are not one to one mapping with elements in a manufacturing process. The notions of order and timing of events are usually not explicit in the OR models and the execution of the optimization is typically a black box to the user. This makes debugging of an optimization model a challenging task. Also, OR models typically lack the modularity of modern object-oriented languages, so they tend to become difficult to modify or extend (Brodsky and Nash 2005).

On the other hand, simulation modeling is useful in cases where the manufacturing problem is too complex to be represented as an objective function and constraints using mathematical equations and many parameters may be unknown. Law and Kelton define simulation as "the process of constructing a model of a system that contains a problem and conducting experiments with the model for the specific purpose of solving the problem and aiding in decision-making (Law and Kelton 2000)." The elements of a simulation are state variables and transitions; depending on how one models the problem, it may have a clear one to one correspondence with elements of a manufacturing process, e.g., a machining module in a simulation corresponds with a machining center in a real factory. Real-world time and sequence of events correspond to time and sequence in simulation runs, so it is easy to understand and debug. Also, object-oriented software allows modular development of simulation models. Previous studies have been conducted to support decision-making for SM using modeling and simulation. Heilala et al. (2008) described an interactive tool that jointly combined simulation models and data to support human- and environment-friendly decision-making for SM systems regarding investment and product design. Solding et al. (2009) described an integrated system that combined simulation and an energy analysis tool, Method for analysis of INDustrial energy systems (MIND) (Tari and Söderström 2002), to optimize energy use in foundry plants. Johansson et al. (2007) presented a simulation model of an automotive paint shop that simulated different input parameter options to determine the one with least CO_2 emission. Berglund, et al. (2011) developed a simulation model for a precision casting operation to study issues associated with integrating production system, process energy, and facility energy in order to improve manufacturing sustainability. However, in our analysis, none of these research approaches follows a systematic methodology that uses reusable modular knowledge description of the sustainable process and analytics for analysis and decision-making.

Decision guidance methodologies Compared to simulation, OR modeling has a major advantage. If modeled correctly using a manageable constraint domain such as Linear Programming (LP) or Mixed Integer Linear Programming (MILP), an optimization problem can be solved efficiently using solvers with sophisticated optimization algorithms. Brodsky and Nash (2005) developed CoJava, which has both the advantages of simulation-like (i.e., modular model that provide one to one mapping to the problem) modeling and the capabilities of decision optimization. The syntax of CoJava extends Java with special constructs to (1) make a non-deterministic choice of a numeric value, (2) assert a constraint, and (3) designate a program variable as the objective to be optimized. A set of non-deterministic execution paths are defined, each execution path has specific selection of values in the choice statements. A reduction was developed to a standard constraint optimization formulation. A constraint compiler translates it into a similar Java program in which the primitive numeric operators and data types are replaced by symbolic constraint operators and data types. The program is then compiled and executed to produce a symbolic decision problem, which can be solved by an external optimization solver (Brodsky and Nash 2005).

Brodsky and Wang (2008a) developed the decision guidance management system methodology, in which they used a similar concept, but replacing Java programming with relational database modeling. Decision guidance query language (DGQL) (Brodsky and Wang 2008a) allowed users to define objective functions over the augmented table. The optimization semantics of DGQL are to (1) find an optimal non-deterministic query evaluation, i.e., one that produces the minimum or maximum as required and (2) compute the query with values for the non-deterministic attributes corresponding to the optimal evaluation path. MP techniques are used to instantiate the variables (Brodsky et al. 2009). DGQL combines the database application modeling with optimization algorithms. Structured Query Language (SQL) (ISO/IEC 2011) and DGQL commands are used to formulate optimization problems that can be transformed to corresponding mathematical models and solved by a variety of meta-optimization heuristics and commercial optimization solvers such as IBM ILOG CPLEX, AMPL, SNOPT, and MINOS (IBM 2014; AMPL 2014; Gill et al. 2008; AMPL 2011).

CoJava and DGQL have made efforts to unify the modeling of data query, what-if analysis, and decision optimization. However, they do not provide built-in support for manufacturing process modeling and sustainability metrics representation and reuse. The SPAF developed at NIST and used in this paper leverages the concepts and algorithms of CoJava and DGQL to represent manufacturing process knowledge for sustainability analysis and decision-making. SPAF models will be translated to a standard optimization model in Optimization Programming Language (OPL), which can be executed using CPLEX to derive actionable recommendations.

Overview of sustainable process analytics formalism

Manufacturing industries need systematic methodologies to help improve energy and material efficiency, and lower emissions and costs to achieve sustainable manufacturing goals. SPAF supports such a methodology by modeling the metrics, key performance indicators (KPIs), and characteristics developed for unit manufacturing, assembly processes, and production planning to provide performance assessment and recommendations for improvement. This section provides an overview of SPAF. More detailed SPAF syntax, semantics, and examples have been provided in Brodsky et al. (2014).

SPAF is designed for SM knowledge representation, analysis, and optimization to support the sustainable manufacturing decision-guidance methodology. SPAF allows users to formally describe (1) process structure, (2) process data, (3) control variables, and (4) analytical models of sustainability metrics and constraints. The process structure includes the hierarchical composition of processes, sub-processes, and resource flows. Process data include production- and sustainability- related information, attributes, and parameters. Control variables can be instantiated by both users and the systems that implement the formalism. Process analytics expression includes mathematical specification for metrics, constraints, and objectives. Similar to the research idea in (Peng et al. 2014), SPAF supports modular, extensible, and reusable models. It allows users to model underlying reality once and reuse it for different tasks. SPAF includes three major parts.

(1) Generic analytics language—provides unified syntax and semantics for process modeling and support of data querying, what-if analysis, and decision optimization.
(2) Process description and sustainability metrics—enable formal representation of process structure, resource flow, data, control variables, objectives, and constraints; support sustainable metrics models; and are easy to use by manufacturing and business users on a factory floor.
(3) Reduction procedure—enables the translation of SPAF queries into specialized models such as optimization models to generate actionable recommendations for SM improvement.

SPAF features summary Table 1 shows the comparison of SPAF with some of the other modeling languages. The modular design of SPAF enables the built-in process modeling and the creation of a model library that in turn supports sustainability metrics definition and reuse. Reuse makes it particularly easier for manufacturing and business users to model processes, especially if there are graphical user interfaces available.

Decision guidance methodology

This section presents a model-based decision guidance methodology that provides procedures and guidelines for systematic sustainability analysis based on SPAF. Potential users of a decision guidance management system that apply the methodology are also introduced. As discussed in the introduction section, one of the challenges to use optimization techniques is that complex analysis requires the development of formal simulation and/or optimization models, which often requires significant modeling expertise and substantial development effort. For example, modeling of a manufacturing process using Mathematical Programming (MP) or Constraint Programming (CP) tools typically requires extensive OR and mathematical training, which normally people from the production floor do not have. The SPAF makes process analysis modeling more intuitive and straight forward for domain users such as process engineers. Figure 1 shows the concept and potential users of such a system. There are three types of potential users, each with different roles in using and/or maintaining the system.

- SPAF *analyst* The analyst is a primary user of the system. S/he may be an engineer who is in charge of defining case scenarios; setting up analysis model objectives together with decision makers; identifying sustainability metrics, constraints, and controls; and performing model development and execution. S/he will collect data from the factory floor and represent process structure, flow, and metrics using SPAF.
- *Decision maker* The decision maker is an end user of the system. S/he identifies the SM objectives/goals and provides model requirements. The decision maker will query the knowledge base, ask what-if analysis questions, and/or make optimization requests with applied constraints and control data for a specific problem. S/he may also quickly compose SPAF models simply using model components in a SPAF library.
- SPAF *knowledge-base administrator* The knowledge-base administrator serves as a system administrator who is responsible for updating system data, creating reusable knowledge artifacts, helping other users develop new applications, maintaining the SPAF model library, modifying/improving the system design, and maintaining/enhancing the system.

Figure 2 shows the steps of the methodology that guides users discussed above to model and perform sustainability analysis and derive decision recommendations using SPAF. The methodology involves: (1) identifying manufacturing process scenarios, in which the sustainability performance need to be assessed, analyzed, and optimized; (2) collecting domain knowledge for the relevant indicators and their metrics of the scenario; (3) developing conceptual models of the manufacturing processes; (4) collecting and processing relevant data that required for the modeling and study; (5) modeling the data and the scenario using SPAF; (6) performing what-if analysis and decision optimization; and (7) generating actionable recommendations for users. These steps are discussed in the following subsections.

Formulation of high level problem

More and more manufacturers consider sustainability in different aspects such as product design, production and

Table 1 A comparison table of SPAF and other languages

Features	Modeling languages					
	Process description languages (PSL, BPMN, SysML)	Database query languages (SQL, XQuery)	Simulation languages (SIMAN, OO languages)	Optimization modeling languages (AMPL, GAMS, OPL)	Optimization semantics for OO and query Lang's (CoJava, DGQL)	Design goal for SPAF
Data manipulation and querying	Do not directly support	Supports	Require modeling and programming	AMPL and GAMS are not designed for query processing; OPL has some built-in support	Supports	Supports
What-if analysis	Do not directly support	Limited (only what can be expressed as DB queries)	Supports	Do not support	Supports	Supports
Optimization	Do not support	Limited and not efficient	Limited and not efficient	Supports	Supports	Supports
Unified modeling for different tasks	Do not support	Do not support	Do not support	Do not support	Supports	Supports
Built-in support for process modeling and sustainability metrics	Can be extended to support	Can be extended to support	Can be built on top	Do not support	Can be extended to support	Supports if with a components library
Modular, extensible, and reusable	Supports	Do not support OO extensibility	Supports	Difficult to reuse models	CoJava—support; DGQL—based on SQL	Supports if with a components library
Ease of use (by manufacturing and business users)	Can be easy via graphical interface	Relatively easy (SQL skills required)	Programming skills to model analytics; many allow high-level composition functionality	Math/optimization modeling skills required	CoJava (programming skills required); DGQL (SQL skills required)	Easy for composite process, especially if a graphical interface is added; similar to OPL for atomic process models

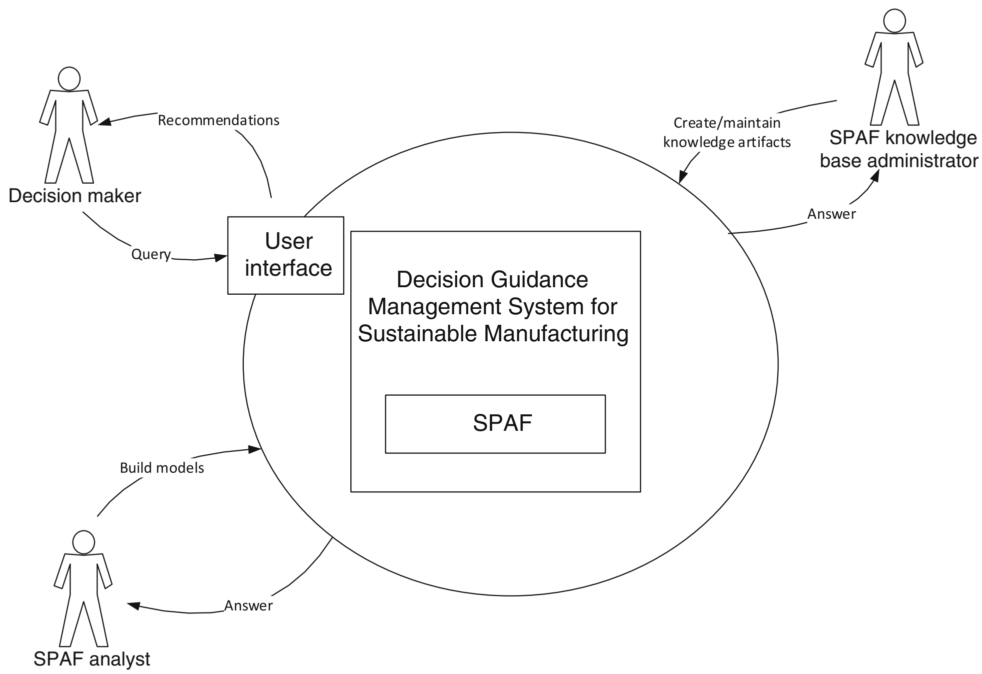


Fig. 1 Concept and users of a decision guidance management system for sustainable manufacturing using SPAF

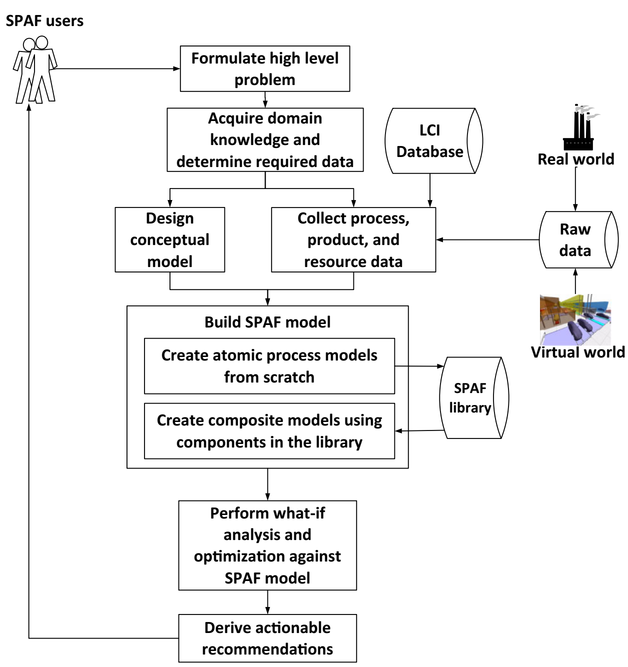


Fig. 2 A model based decision guidance methodology for sustainable manufacturing

investment planning, and corporate sustainability objectives to satisfy multiple stakeholder requirements. They need to assess their sustainability performance and take actions to improve it. To formulate a problem for sustainability analysis using SPAF, the decision maker(s) and SPAF analyst(s) need to follow the steps to:

- Identify or define a manufacturing process, e.g., an assembly process.
- Define objective(s) of the study, e.g., seeking the best process plan that results in minimal energy consumption for producing a product.
- Define the scope of the study including assumptions, conditions, and level of abstractions for the problem.
- Determine the types of information required for decision-making, e.g., the optimal cutting speed range of a machining process for minimal energy consumption.

Once the problem is identified or defined, the domain knowledge and relevant information need to be collected.

Acquisition of domain knowledge

To model and analyze sustainability performance of manufacturing processes, both manufacturing- and sustainability-related knowledge and information need to be collected. As such, modelers need to:

- Understand the nature of the problem and requirements of the study, for example, the analysis of a machining process may require information about characterization of the process, material, cutting tool, and finished parts; machining conditions; tool requirements and tool paths; coolant and lubricant requirements; waste restrictions; and energy used.

- Identify KPIs, which include both sustainability and conventional indicators, e.g., energy and material efficiency, waste, emission, cost, cycle time, throughput, and quality.
- Identify the relationships among the indicators, e.g., equations that convert energy units to equivalent carbon dioxide emissions.
- Identify sustainability metrics, controls, and constraints, e.g., energy consumption as a metric, machining time as a control variable, and spindle speed range as constraints.
- Identify or/and define mathematical models for required metrics computation, e.g., a formula to calculate the emissions from a machining process.

Having knowledge is not enough; it needs to be fit into a big picture to prepare the analysis and provide decision guidance for SM.

Design of problem conceptual model

A conceptual model is a simplified representation of a real world problem. It provides the right level of abstraction that satisfies the modeling objectives and focuses on the metrics of concerns. It can be regarded as sketches of the problem based on the knowledge collected; a conceptual model helps modelers better understand the problem and prepare for solutions. It is a necessary step that helps a modeler think, sort, and solve the problem. When designing a conceptual model that will be implemented using SPAF, the following typical questions need to be answered by the designers. The questions help users abstract the problem and plan the conceptual modeling:

- What are the processes and sub-processes that need to be modeled?
- What are the inputs and outputs of each process?
- What are the relationships between processes?
- What are the indicators and how do they cascade down?
- What are the metrics of each indicator?
- What are the data requirements for the metrics?
- What are the dependency relationships (e.g., correlations) among the indicators and inputs?

The conceptual models provide further detail requirements for data collection.

Collection of data

Data are crucial for performing successful analysis and providing meaningful decision guidance, Skoog's analysis of discrete event simulation (DES) projects shows that about 31 % of the total project time is used for gathering, extracting, and processing data (Skoog 2009). Specific data content, units, and formats are required during model development and execution. For example, life cycle assessment (LCA) studies involve data from life cycle inventory (LCI) databases, what-if analysis needs different sets of input data for computation and comparison for alternatives, and optimization requires data for parameters of closed-form mathematical expressions. Therefore, the sources of data including sustainability information need to be identified; it can be from LCI databases, the real world, or a virtual world. Currently, in most cases, sustainability-related data are not being identified and collected (Shao et al. 2010). Even in cases where these data are available, they are often stored in various forms. Some data need to be aggregated or decomposed; new metrics may be only obtained through computations. For example, total energy consumption is calculated as the sum of all energy consumed by different processes within the factory floor and CO_2 emissions is derived from energy consumption. What information is available needs to be determined, for those data that are not available but required, estimation or approximation using various mathematical techniques such as the Least-Squares estimation method may need to be applied (Brodsky et al. 2008b; Shao et al. 2009). There may be historic, experimental, and statistical data that can be leveraged to learn unknown parameters. Data collected have to be converted to the correct units in order to have meaningful results, for example, if second is used for time, all data units for time need to be in seconds. Shao et al. (2010) have discussed several data sources and data collecting methods including LCA methodology and LCI data, real world data, and virtual world data. The domain knowledge and data collected provide foundations required for SPAF modeling.

Building SPAF models

In this subsection, approaches and procedures for different users to model problems using SPAF are discussed. "Formulation of high level problem" section to "Collection of data" section have prepared users with the knowledge to answer the following questions. The answers to these questions enable users to have clear goals of the model, define structure of the model, and accurately model the problem using SPAF.

- What are the objectives of the model?
- How many processes need to be modeled?
- Is a process a composite process that has sub-process(es) in it or an atomic process that is an end process in which there is no sub-process?
- What are the items that flow into and out of the processes?
- What are the metrics that need to be calculated in each process?
- What are the overall sustainability metrics for the composite process?

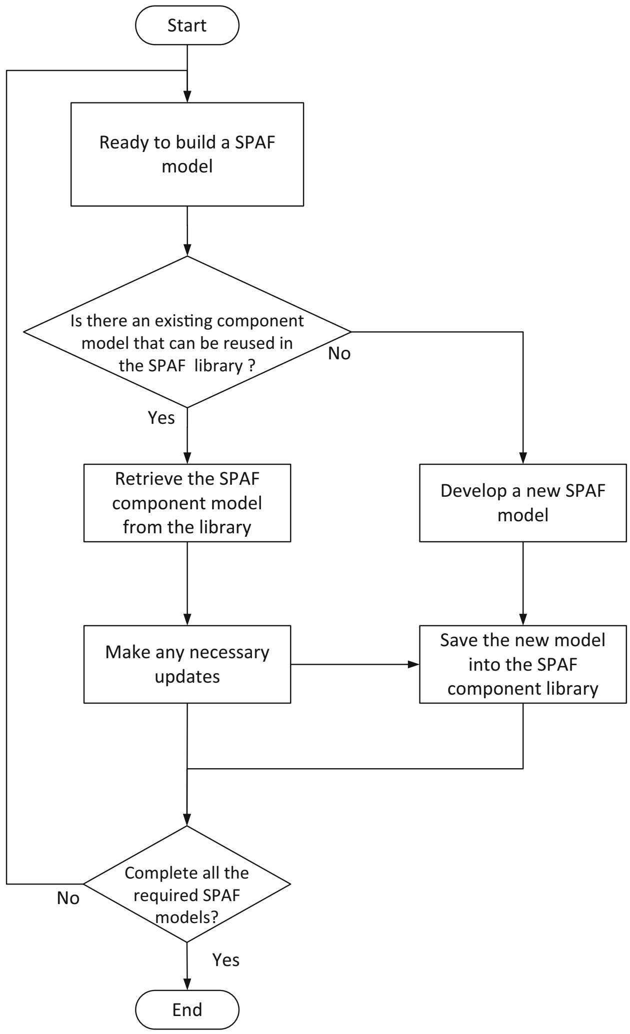


Fig. 3 SPAF model building approach and procedure

- What are the mathematical equations required for modeling the process?
- What are the explicit data, i.e., what data are available?
- What are the implicit data, i.e., what data are not directly available but can be computed?
- What are the data types?
- What are the data needed to be determined (decision variables), e.g., what are the data that can be controlled by the users or instantiated by an optimization solver?
- What are the constraints of the model, i.e., conditions that need to be satisfied?

Users follow the flowchart in Fig. 3 to decide the approach and procedure they need to use. This is a key feature of this methodology.

According to the knowledge collected, the conceptual model developed, and the questions answered; users can develop modular SPAF sub-models that include context models, flow models, flow aggregator models, atomic process models, and sustainability metrics computational models. A composite process model integrates all related sub-models by using *include* statements. As indicated in Fig. 2 and reflected in Fig. 3, there are two kinds of modeling scenarios:

(1) If there is no appropriate model component in the SPAF library to adopt, modelers need to create new SPAF model components and composite models. This requires modelers to have more modeling knowledge and/or training. The created model components can then be stored in the SPAF library for future use.
(2) If there are appropriate model components available in the SPAF model library, the modelers can create composite models by including the model components. Note that the modelers need to ensure that new data are used and model modification might be needed. Modifying existing model components is much easier than developing new models. This type of modeler might be decision makers who only require minimal training on SPAF modeling.

According to the syntax and semantics defined in Brodsky et al. (2014), the typical procedures of creating SPAF models are explained briefly as follows. A few high level flowcharts are also shown.

- To create a SPAF *context* model, the model name needs to be defined first and then data are declared, the data can be accessed by other related SPAF models.
- To create a SPAF *flow* model, a parameter *Id* needs to be defined, it will be replaced by the actual parameter in an *include* statement that calls the flow model. The context model needs to be included. Data types and structures for the flow need to be declared. Data constraints, if any, should be defined.
- To create a SPAF *flow aggregator* model, as the flow model, a parameter *Id* needs to be defined and will be replaced later by the actual parameter in an *include* statement that calls the flow aggregator model. The context model needs to be included. Data types and structures for both input and output flows need to be declared. Data constraints, if any, should be defined. Normally, one of the constraints is that the total number (amount) of the input flows equals to the total number (amount) of the output flows.

Figure 4 is the flowchart for creating a generic atomic process model. The *Id* is provided when it is called. For every flow, input flow or output flow, the flow model is *included* with a parameter of the flow name. Metrics data type and structures are declared. Metrics computations are expressed. Necessary constraints are also presented in the model.

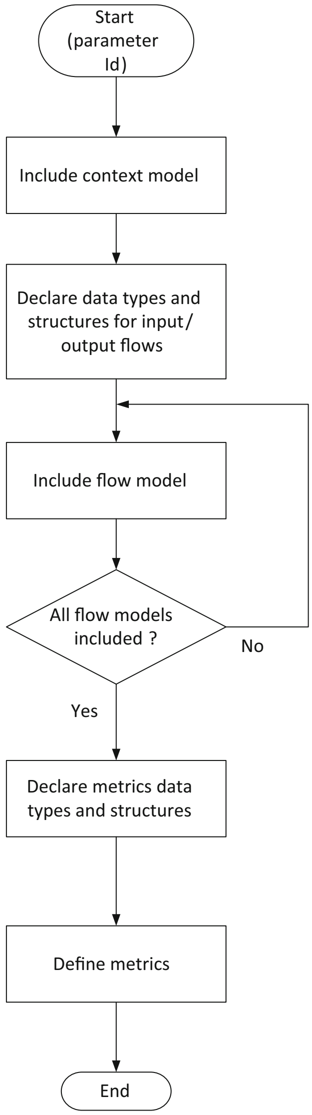
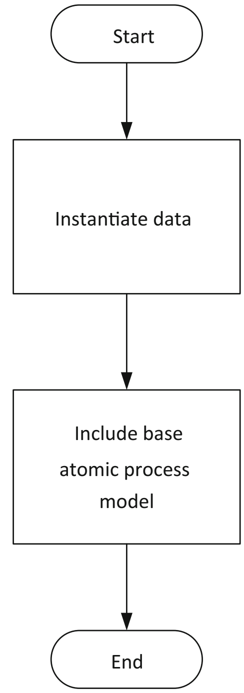


Fig. 4 Generic atomic model flowchart

Figure 5 is the flowchart for an atomic process instance model. First, data defined in the generic atomic model are instantiated; this is the place where instance data are used to replace the three dots and integrate the two models. In SPAF, the three dots "…" is used to express the missing data that need to be instantiated as a constant, or an expression before the data is used. The generic atomic process model is then included with the parameter *Id* that is replaced with an actual parameter, e.g., a machine name.

Figure 6 is the flowchart for creating a composite process model. Model *Id* will be replaced when an SPAF query is formulated. It includes the context model. All the sub-processes are declared and included by calling the process models with appropriate parameters. Then, each flow and flow aggregator model are included. Finally, the computations of aggregated metrics are expressed. Sustainability metrics models

Fig. 5 Atomic process instance model flowchart



are computational models for specific problems. Once built, they can be stored in a model library for reuse in the future.

By following these flowcharts, modular SPAF models such as context models, flow models, flow aggregator models, atomic process models, and composite process models can be incrementally developed without changing any of the existing models or queries. This is in contrast to the current common practice where the introduction of a new component or factor leads to major redesign and reimplementation of the models and associated algorithms for a decision optimization task.

What-if analysis and optimization query using SPAF

With all the required SPAF models developed, a SPAF analytical query can be formulated. There are mainly two types of queries: What-if analysis query and decision optimization query. Figure 7 is the flowchart for creating a SPAF query. First, data required by the SPAF models are included. Then, the complete model of a composite process is included. Constraints, if any, are added. Finally, if it is an optimization query, depending on the optimization objective, an optimization statement (minimize or maximize) needs to be added. Note that a what-if analysis query does not need to have the optimization statement.

Using a translator/compiler that implements the SPAF query computation algorithms, the analytical query formulated by users can be translated to a standard optimization model such as an OPL model, which can be solved by using an optimization solver. If the problem is valid and feasible, an optimal solution will be provided as actionable

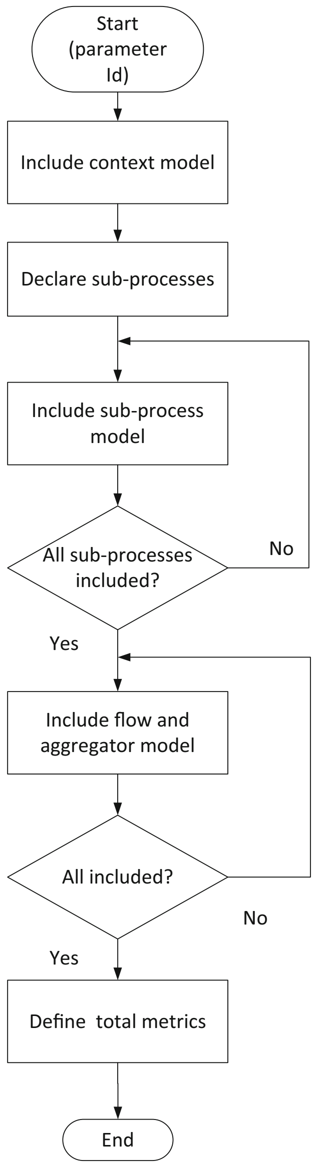


Fig. 6 Composite process model flowchart

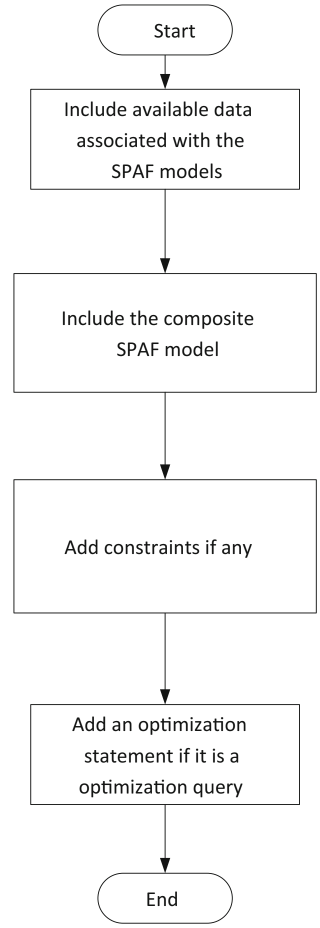


Fig. 7 SPAF query model flowchart

recommendations to decision makers. Figure 8 depicts these steps of solving an analytical query in a decision guidance management system.

Query type, what-if analysis or decision optimization, is decided based on the nature of the problem. What-if analysis is a deterministic computational model that allows users to try a wide variety of scenarios using the same model with different sets of input data such as historical process, production, and sustainability data to predict different outcomes. The query can then be translated to an explicit analytical sequence, and all variables in the expressions are instantiated through computation using data provided. Different answers of what-if analysis allow decision makers to compare and assess changes before they happen. For example, given certain settings of a process's controls and input, the total energy consumption, carbon emission, and costs can be predicted before investing on a machine tool. In an optimization query, the optimization objectives are indicated using *minimize* or *maximize* statements. Some data are not explicitly given, they will be treated as decision variables, which will be instantiated by an optimization solver using values that satisfy all the constraints and the objectives.

Actionable recommendation

SPAF models enable knowledge representation of manufacturing processes and compute relevant sustainability metrics. A decision guidance management system that is based on SPAF models generates the SPAF query results, i.e., actionable recommendations that help decision making for SM performance improvement. For example, investment decision of a machining center with optimal parameters requirement, process parameter settings decisions are made to achieve the minimal energy consumption of the process. According to the decision recommendations, users can determine the

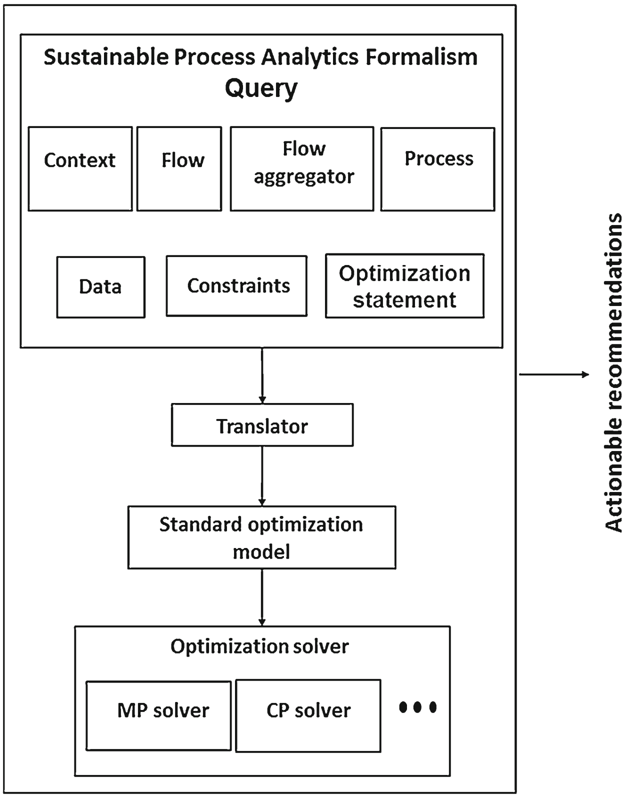


Fig. 8 Steps for solving an SPAF query in a decision guidance management system

material, labor, equipment, tooling, inventory, material handling, process planning, and maintenance requirements to manage energy, emission, and waste systematically. The recommendations are the results of the two kinds of queries:

- What-if analysis—query results from explicit data for different scenarios that can be computed and compared to select the most appropriate option.
- Decision optimization—query results from decision optimization that are the best alternative that satisfies users' objectives and constraints.

The systematic analysis and quantitative recommendations not only provide decision makers with confidence, but also give them flexibility to try different options, e.g., by defining a new scenario or changing input data, modifying sustainability objectives, and adding/removing constraints.

Case study

A manufacturing case study has been performed to demonstrate the application of the SPAF-based decision guidance methodology on two production levels. The case study is designed to make optimal decisions for a production process and a unit-process. The SPAF methodology is used first to present how to allocate resources for minimal energy consumption at the production level and then to present how to select process parameters for minimal energy consumption at the unit-process level. The description of the sub-sections follows the procedure of the SPAF-based decision guidance methodology as shown in Fig. 2.

Problem formulation

Figure 9 presents an example of a production process to manufacture an assembly part. A manufacturer produces a mechanical part for a drilling tool. Two metal parts are joined in a welding process and plastic grains are used in an injection molding process. Metal powders are used in a die casting process and a cast part flows into one of three machine tools for turning process. All three parts are then fastened together by human laborers at a threaded fastening station. The machine tools can produce a maximum of 20 parts per hour. So this production process will manufacture 20 assembled parts per hour. Parts can be allocated according to human experience or decision recommendation from a system. For this problem, the goal is to minimize the total energy consumption of the overall process.

The objectives of this case study are: (1) to minimize energy consumption in the production process by properly allocating the number of parts to each of the three machine tools and (2) to minimize energy consumption of a turning machine tool (for example, Turning Machine 1) by appropriately determining process parameters (i.e., feedrate, cutting depth, and spindle speed).

Knowledge acquisition and data determination

Production level: For the production plan optimization, process engineers would like to find and deploy the best production plan that satisfies demands and minimizes the total energy consumption in kJ. An optimization model is used to instantiate optimal operational parameters and control variables that satisfy all the constraints. These optimal parameters include the optimal settings for the allocation of parts flowing into the three turning machines, which are set as decision variables. The objective function is defined as 'minimize energy consumption' where 'energy consumption' is a performance metric. Constraints include 'hourly throughput of the production process' and 'maximum capacity of the machine tools'. Decision variables are 'the numbers of parts allocated to the three machine tools', as described in the previous sub-section. We assume that energy consumption has a piecewise linear dependency on the number of parts. Similarly, the energy consumptions of other unit-processes also have piecewise linear dependences. Equation 1 expresses the piecewise linear formula that is used to calculate energy

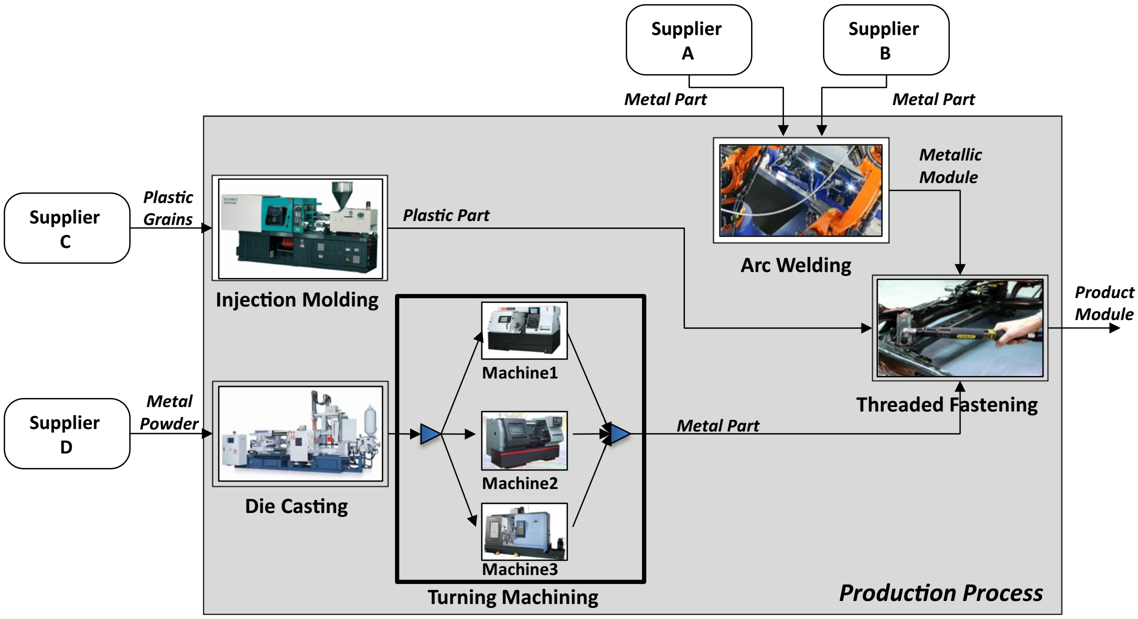


Fig. 9 An example of a production process

consumption in the unit-processes. The three machine tools will have different energy patterns that depend on their energy performances.

$$E(x) = \begin{cases} \alpha_0 x + \beta_0, & if \quad 0 \le x < x_1 \\ \ldots & \\ \alpha_{n-1} x + \beta_{n-1}, & if \quad x_{n-1} \le x \le x_n \end{cases} \quad (1)$$

where, x: the number of parts, E(x): energy consumption, α: gradient coefficient, β: Y intercept.

Unit-process level: For the unit-process optimization, the objective of the model is to obtain minimal energy consumption and the corresponding optimal values of the decision variables, i.e., the objective function is defined as 'minimize energy consumption'. Because energy consumption can be obtained by multiplying power consumption with machining time in a turning process, power consumption needs to be characterized in terms of the machine tool's status. The power consumption has different values for different operational mode such as setup, idle, active, or teardown. In the setup, idle, and teardown modes, machines consume constant power, which is determined by the machine tool's performances. While in the active mode, power consumption of a process depends on the settings of the parameters such as feedrate, spindle speed, and cutting depth, because the process parameters' influence on cutting power during actual machining. Therefore, decision variables for the problem are 'cutting depth', 'spindle speed' and 'feedrate'. Constraints are 'ranges of the three process parameters' because the parameters should be selected within feasible cutting conditions.

We assume that power consumption of Turning Machine 1 can be predicted by an empirical model, which is derived by experiment and measurement. Equation 2 presents such an empirical model for power consumption prediction with respect to a number of process parameters, there are total three process parameters in this case, see Kim et al. (2014) for details.

$$P(x) = \gamma_0 + \sum_{i=1}^{n} \gamma_i x_i + \sum_{i \le j}^{n} \gamma_{ij} x_i x_j + \varepsilon \quad (2)$$

where, x: decision variables (cutting depth, spindle speed, and feedrate), P(x): power consumption, γ: coefficient, ε: error, n: the number of decision variables

Data collection

Production level: it is necessary to collect the data relevant to energy consumption because we need to calculate the actual energy consumed by the production process. In other words, Eq. (1) needs to be instantiated with real world data. Figure 10 visualizes the line graphs that represent the energy equations assumed for the three machine tools. Energy patterns for other unit-processes are also collected to calculate energy consumed.

Unit-process level: similarly, an experiment has been conducted for deriving the power equation of Turning Machine 1. A power meter measures time-series power consumed at Turning Machine 1 under a Box–Behnken experimental design. In this experiment, the constraint of 'feedrate' is given a range of (0.23–0.27) mm/rev, 'spindle speed' in a range of (100–166.5) rad/s, and 'cutting depth' in a range of (2–3) mm, respectively. A quadratic regression fitting method is used to generate the empirical model for power consumption

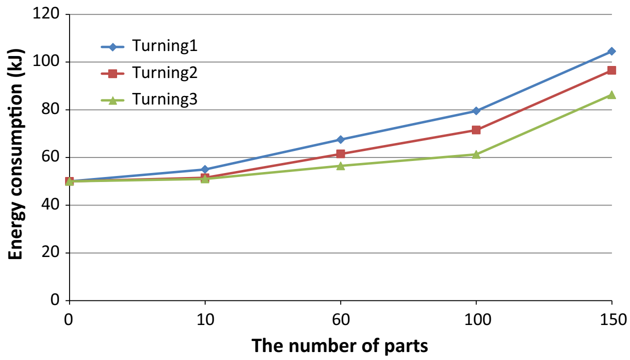


Fig. 10 Energy patterns of the three machine tools

during the active mode (the detail of the experiment is out-of-scope of this paper). Equation 3 presents the regression model derived from the experiment. To validate the regression model, we should have enough measurement data sets for both the estimated and real machine power consumptions. Then the percentage differences between the estimated power consumption and the real machine power consumption are used to indicate the accuracy of the model, i.e., the smaller the percentage difference is, the more accurate the model is. In this simple case, we have used all the fifteen experiment trial parameter values assigned by the Box–Behnken design for the regression model generation; here we randomly pick one more parameter set (feedrate = 0.265 mm/rev, spindle speed = 166.5 rad/s, and cutting depth = 3 mm) as an example. The result indicates that the estimated power consumption is 3.93 kW while the real-machined power consumption is 4.10 kW, so for this example, the percentage difference is 4.23. Equation 3 is case specific, because the coefficients can vary for different machine tools, workpiece materials, cutting tools, machining operations, and machined parts.

$$
\begin{aligned}
P(x) = {} & 2.807 + 0.123x_1 + 0.676x_2 + 0.157x_3 \\
& - 0.028x_1^2 + 0.053x_2^2 - 0.008x_3^2 \\
& + 0.176x_1x_2 + 0.022x_1x_3 + 0.024x_2x_3 \qquad (3) \\
x_1 = {} & \frac{X_1 - 0.25}{0.02},\ x_2 = \frac{X_2 - 133.25}{33.25},\ x_3 = \frac{X_3 - 2.5}{0.5}
\end{aligned}
$$

where, P(x) = power, X_1 = federate (mm/rev), X_2 = spindle speed (rad/s), X_3 = cutting depth (mm).

SPAF modeling

For the purpose of SPAF modeling, Fig. 9 is redrawn as Fig. 11 to indicate the flow, flow aggregators, the metrics, etc. The SPAF models are developed to describe processes shown in Fig. 11, their input data, KPIs, metrics, optimization objective, and constraints for the defined two case scenarios. Since there is no existing SPAF model for these problems, all the models have to be created from scratch.

Production level: The SPAF models are developed to represent the flow model, the flow aggregator model, the seven atomic process (rectangles in Fig. 11) models, a composite model, and the computational models of energy consumption. The SPAF modules are modeled according to the syntax and semantics of SPAF (Brodsky et al. 2014). Figure 12 shows a section of the SPAF model. The 'model flow item {}' indicates the flow model and the 'model flow aggregator itemAggr {}' stands for the flow aggregator model. The 'model process baseEnergyThruMachine {}' represents the function of energy consumption, based on a piecewise linear function 'pwlFunction energyFunction' and its constraints. 'Turning 1' is an example of a turning process model. The 'turning 1' includes the textual code for the piecewise energy pattern, as shown in Fig. 10. The 'model process Manufactur-

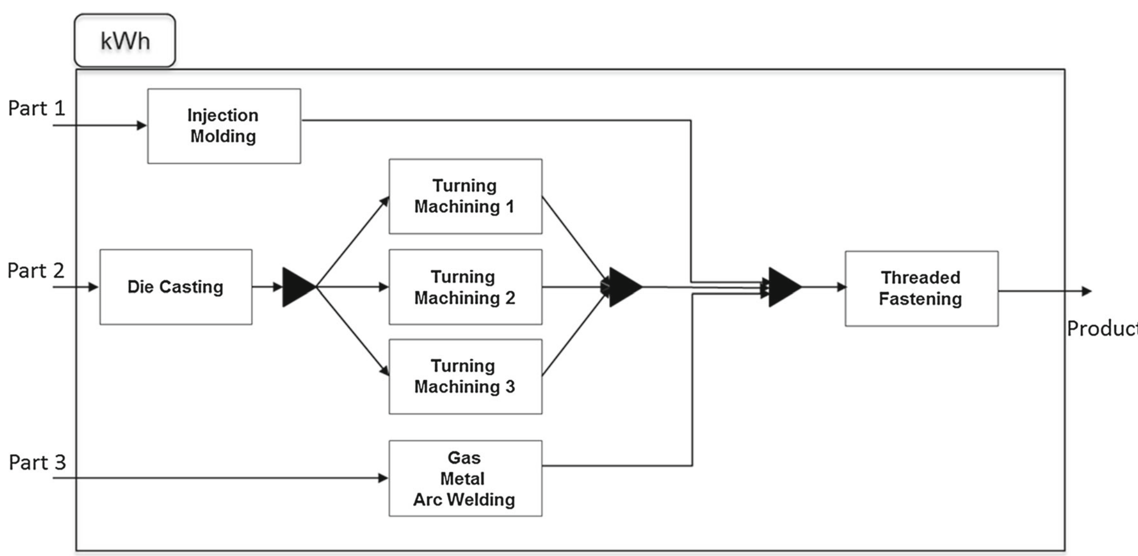


Fig. 11 SPAF flow diagram for the case study

```
model flow item {
            string type = ...;    // unique id for every part in production
            float unitPerHour; }
model flow aggregator itemAggr {
            ...
            sum(i in inputFlows) item[i].unitPerHour == sum(j in outputFlows) item[j].unitPerHour; }
model process baseEnergyThruMachine {
            ...
            forall(i in inputFlows)
                        forall(o in outputFlows)
                                    item[i].unitPerHour == item[o].unitPerHour * inputPerOutput[i];
            ...
            pwlFunction energyFunction = piecewise{s[1] -> b[1]; s[2] -> b[2]; s[3] -> b[3]; s[4]}(minThru, initEnergy);

            float thru;
            minThru <= thru <= maxThru;
            float energyPerHour = energyFunction(thru);

            forall (i in outputFlows) thru == item[i].unitPerHour; }
model process Turning1 {
            ...
            model energy = new baseEnergyThruMachine["turn1"]{
                                                            inputFlows = { turningInItem1 };
                                                            outputFlow = turningOutItem1;
                                                            inputPerOutput[] = [1];
                                                            s[] = [0.2, 0.25, 0.30, 0.5];
                                                            b[] = [10,60,100];
                                                            minThru = 0.0;
                                                            maxThru = 150;
                                                            initEnergy = 50.0};
            float energyPerHour = baseEnergyThruMachine[energy].energyPerHour;
            float thru = baseEnergyThruMachine[energy].thru; }
model process ManufacturingFloor {
            ...
            float energyPerHour = DieCasting[dieCastingProc].energyPerHour + Turning1[turning1Proc].energyPerHour +
                        Turning2[turning2Proc].energyPerHour + Turning3[turning3Proc].energyPerHour +
                        GasMetalArcWelding[gasMetalArcWeldingProc].energyPerHour + InjectionMolding[injectionMoldingProc].energyPerHour;

            item[threadedOutItem].unitPerHour == 20; }
minimize ManufacturingFloor[].energyPerHour;
```

Fig. 12 An SPAF sample model for the resource allocation problem

ingFloor { }' represents the problem formulation that defines the total energy consumption, the constraint, and the objective function.

In this SPAF model, the 'turning 1' process model can be reused to define the process models for 'turning 2' and 'turning 3' because the models have the same type and similar data content with different instantiations.

Unit-process level: Fig. 13 expresses a partial SPAF model for the unit-process level. 'Model process TurningMachineOrders' defines the constraints of the three process parameters, the active power (P_active[o]), the real machining time (T_active[o]), and energy equation (E_active). 'Model process TurningMachine' identifies real values of the constraints. The objective function is represented as 'minimize TurningMachine [].Energy_Consumption' at the end of the program.

Optimization solutions

Production level: an optimal solution can be derived through the steps of solving an SPAF model-based query as explained in Fig. 8. In this case study, IBM CPLEX is chosen as the optimization solver, which executes OPL models as an input. A SPAF model is then translated to two OPL files (i.e., a model file and a data file) through a translator as shown in Fig. 8. The translator forms a structure of data tokens on the basis of syntax and semantics of the SPAF model, and then it converts the SPAF-based data structure to an OPL-based

```
model process turningMachineOrders {
...
Min_Depth_Cut  <= Depth_Cut <= Max_Depth_Cut;
Min_Spindle_Speed  <= Spindle_Speed  <= Max_Spindle_Speed;
Min_Feed_Rate  <= Feed_Rate  <= Max_Feed_Rate;
...
float P_active[o in Orders] = 2.807 + 0.123*(Feed_Rate[o])  + 0.676*Spindle_Speed[o]  + 0.157*(Depth_Cut[o])  - 0.028*(Feed_Rate[o])*(Feed_Rate[o])
                    + 0.053*Spindle_Speed[o]*Spindle_Speed[o]-0.008*(Depth_Cut[o])*(Depth_Cut[o])  + 0.176*(Feed_Rate[o])*Spindle_Speed[o]
                    + 0.022*(Feed_Rate[o])*(Depth_Cut[o])  + 0.024*(Spindle_Speed[o])*(Depth_Cut[o]);

float T_active[o in Orders] = 60 * o.WorkPiece_Length  * stroke[o]/((Feed_Rate[o])*  Spindle_Speed[o]);
float E_active [o in Orders] = P_active[o]  * T_active[o];
}
model process TurningMachine {
...
Min_Depth_Cut  = 2.0;
Max_Depth_Cut  = 3.0;
Min_Spindle_Speed  = 100.0;
Max_Spindle_Speed  = 166.5;
Min_Feed_Rate  = 0.23;
Max_Feed_Rate  = 0.27;
float Energy_Consumption  = turningMachineOrders[demoTurningMachine].Energy_Consumption;
...
};
minimize TurningMachine[].Energy_Consumption;
```

Fig. 13 An SPAF sample model for the parameter selection problem

```
{string} item_IDS = ...;
string item_type[item_IDS] = ...;
dvar float item_unitPerHour[item_IDS];
...
{string} itemAggr_IDS = ...;
{string} itemAggr_inputFlows[itemAggr_IDS] = ...;
{string} itemAggr_outputFlows[itemAggr_IDS] = ...;
...
dexpr float Turning1_energyPerHour[Turning1_index in Turning1_IDS] = baseEnergyThruMachine_energyPerHour[Turning1_energy];
dexpr float Turning1_thru[Turning1_index in Turning1_IDS] = baseEnergyThruMachine_thru[Turning1_energy];
...
dexpr float ManufacturingFloor_energyPerHour[ManufacturingFloor_index in ManufacturingFloor_IDS] = DieCasting_energyPerHour[ManufacturingFloor_dieCastingProc] +
            Turning1_energyPerHour[ManufacturingFloor_turning1Proc] + Turning2_energyPerHour[ManufacturingFloor_turning2Proc] +
            Turning3_energyPerHour[ManufacturingFloor_turning3Proc] + GasMetalArcWelding_energyPerHour[ManufacturingFloor_gasMetalArcWeldingProc] +
            InjectionMolding_energyPerHour[ManufacturingFloor_injectionMoldingProc];
minimize ManufacturingFloor_energyPerHour["P-01"];
subject to {
 forall(id in baseEnergyThruMachine_IDS)
  forall(baseEnergyThruMachine_i in baseEnergyThruMachine_inputFlows[id])
  forall(baseEnergyThruMachine_o in baseEnergyThruMachine_outputFlows[id])
   item_unitPerHour[baseEnergyThruMachine_i] == item_unitPerHour[baseEnergyThruMachine_o] * baseEnergyThruMachine_inputPerOutput[<id, baseEnergyThruMachine_i>]; };
subject to {
 forall(id in baseEnergyThruMachine_IDS)
 baseEnergyThruMachine_minThru[id] <= baseEnergyThruMachine_thru[id] <= baseEnergyThruMachine_maxThru[id]; };
subject to {
 forall(id in baseEnergyThruMachine_IDS)
  forall(baseEnergyThruMachine_i in baseEnergyThruMachine_outputFlows[id])
   baseEnergyThruMachine_thru[id] == item_unitPerHour[baseEnergyThruMachine_i]; };
subject to {
 forall(id in ManufacturingFloor_IDS)
 item_unitPerHour[ManufacturingFloor_threadedOutItem] == 20; };
```

Fig. 14 An OPL sample model code for the resource allocation problem

data structure under syntax and semantics of the OPL model. Figure 14 shows a sample of an OPL model file and Fig. 15 presents a sample of an OPL data file. For example, 'float energyPerHour' in the SPAF model is converted to "dexpr float ManufacturingFloor_ energyPerHour [..] = …' in the OPL model.

The optimization solver derives an optimal solution through branch-and-bound. The summary of the optimal

```
item_IDS = { "dieIN" "dieOUT" "turnIN1" "turnOUT1" "turnIN2" "turnOUT2" "turnIN3" "turnOUT3" "weld2fast" "weldIN" "inj2fast" "injIN" "fastOUT" "fastIN" };

item_type = #["dieIN" : "dieCastingInType", "dieOUT" : "dieCastingOutType" ,
"turnIN1" : "dieCastingOutType" , "turnOUT1" : "turningOutType" ,
"turnIN2" : "dieCastingOutType" , "turnOUT2" : "turningOutType" ,
"turnIN3" : "dieCastingOutType" , "turnOUT3" : "turningOutType" ,
"weld2fast" : "weldingOutType" , "weldIN" : "turningOutType" ,
"inj2fast" : "injectionMoldingOutType" , "injIN" : "injectionMoldingInType" ,
"fastOUT" : "finalPartType" , "fastIN" : "addFasteningInType" ]#;

itemAggr_IDS = { "dieAggr" "turnAggr" };
itemAggr_inputFlows = #["dieAggr" : {"dieOUT"}, "turnAggr" : {"turnOUT1", "turnOUT2", "turnOUT3"} ]#;
itemAggr_outputFlows = #["dieAggr" : {"turnIN1", "turnIN2", "turnIN3"}, "turnAggr" : {"weldIN"} ]#;

baseEnergyThruMachine_s = [0.2, 0.25, 0.30, 0.5, 0.2, 0.25, 0.30, 0.5 , 0.15, 0.2, 0.25, 0.5 , 0.1, 0.11, 0.12, 0.5 , 0.2, 0.25, 0.30, 0.5 , 0.2, 0.25, 0.30, 0.5 , 0.2, 0.25, 0.30, 0.5 ];

baseEnergyThruMachine_b = [10, 60, 100, 10, 60, 100 , 10, 60, 100 , 10, 60, 100 , 10, 60, 100 , 10, 60, 100 , 11, 60, 100 ];

Turning1_IDS = { "turn1" };
Turning1_turningInItem1 = #["turn1" : "turnIN1"]#;
Turning1_turningOutItem1 = #["turn1" : "turnOUT1"]#;
Turning1_energy = "turn1";
```

Fig. 15 An OPL sample data code for the resource allocation problem

solution is that the production process can minimize energy consumption up to 316.35 kJ when the numbers of parts are allocated to 7 parts for turning machine 1, 3 parts for turning machine 2 and 10 parts for turning machine 3. This optimal solution is provided to decision makers as an actionable recommendation.

Unit-process level: similarly, at the unit-process level, the SPAF model is translated to an OPL model file and an OPL data file. A constraint programming technology is used to solve the turning process problem in combinatorial optimization problems by IBM CPLEX. In particular, the optimizer solves the problem with constraint propagation and search algorithms. Optimal process parameters result in 0.27 mm/rev for feedrate, 100 rad/s for spindle speed and 2.6 mm for cutting depth. The energy consumption of Turning Machine 1 can be minimized to 178.28 kJ with the optimal process parameters.

Recommendation

Production level: the optimal solution is recommended to decision makers. A decision maker needs to check whether the optimal solution satisfies his or her objectives and constraints. In this case, it is obvious that more parts should be allocated to the turning machine 3 because it is the most energy-efficient one among the three according to Fig. 10. Also, because the allocated number of parts is within maximum capacity of the three machine tools, it is feasible to apply these numbers to the production process. If the decision maker follows the recommendations for allocating the number of parts going into each turning machine, minimal energy consumption should be guaranteed.

Unit-process level: the optimal process parameters for Turning Machine 1 are also recommended to decision makers. The decision makers should follow the recommendations to have minimal energy consumption. Actual measurement of the power consumptions will help validate the solutions.

Conclusion

The key research contributions of this paper include:

(1) Developing a sustainable manufacturing-decision-guidance methodology that provides step-by-step procedures for sustainability performance analysis and decision guidance using the SPAF for SM.
(2) Performing a manufacturing case study at two levels, i.e., production process level and unit process level, to demonstrate the developed formalism and proposed methodology.

Future research may include (1) designing graphical user interfaces (GUIs) to enable drag-and-drop process modeling and push-button solutions, (2) creating common models for SM indicators and metrics, (3) engaging industry to further validate SPAF and identifying new requirements for enhancing SPAF, and (4) standardizing SPAF in an accredited Standard Development Organization.

A GUI serves the one and the only one connecting point between the users and the system. It allows users to formulate, experiment, analyze, and optimize the model simply; and get actionable recommendations from the system automatically. The interface enables users intuitively using domain specific language to perform the modeling tasks. Also it provides the foundation for automating the decision support process. Common model components in a specific domain model library enable the modularity, reusability, extendibility, and composbility. The integration of the model library and the GUI allows users to quickly model manufacturing problems by simple drags and drops.

The main challenge of implementing SPAF and this methodology for real world industry cases is that it is not a standard yet and it has not been widely known and accepted. That is why that the methodology and SPAF need to be further tested and validated through real industrial case studies. Based on the feedbacks, enhancement and extension may be necessary. The standardization of SPAF will be the foundation for system integration and process analysis, and provide affordable solution to manufacturers, especially SMEs for performing the analysis and optimization of their operations and make better decisions towards their sustainable manufacturing goals.

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

Acknowledgements The authors thank NIST Sustainable Manufacturing program testbed project team for the test case discussion, especially Program Manager, Sudarsan Rachuri, for his valuable input and Abdullah Alrazgan, a graduate student from George Mason University, for his effort on SPAF compiler development. The work represented here was partially funded through cooperative agreement #70NANB12H277 between George Mason University and NIST.

References

AMPL. (2011). Using AMPL/MINOS. White paper. Stanford Business Software Inc. http://www.ampl.com/BOOKLETS/ampl-minos.pdf. Accessed 29 June 2014.

AMPL. (2014). AMPL: A mathematical programming language. Web page. AMPL optimization. http://www.ampl.com. Accessed 19 July 2014.

Berglund, J. K., Michaloski, J. L., Leong, S. K., Shao, G., Riddick, F. H., Arinez, J., & Biller, S. (2011). Energy efficiency analysis for a casting production system. In *Proceedings of 2011 winter simulation conference*, (pp. 1060–1071). Phoenix, USA.

Brodsky, A., & Nash, H. (2005). CoJava: A unified language for simulation and optimization. Principles and practice of constraint programming-CP 2005. *Lecture Notes in Computer Science*, *3709*, 877. doi:10.1007/11564751_115.

Brodsky, A., & Wang, S. X. (2008a). Decision-guidance management system (DGMS): Seamless integration of data acquisition, leaning, prediction, and optimization. In *Proceedings of 41st annual Hawaii international conference on system sciences* (pp. 71–81). Waikoloa, USA.

Brodsky, A., Luo, J., & Nash, H. (2008b). CoReJava: Learning functions expressed as object-oriented programs. In *Proceedings of international conference on machine learning and applications forum* (pp. 368–375). San Diego, USA.

Brodsky, A., Bhot, M., Chandrashekar, M., Egge, N. E., & Wang, X. S. (2009). A decisions query language (DQL): High-level abstraction for mathematical programming over databases. In *Proceedings of the 2009 ACM SIGMOD international conference on management of data* (pp. 1059–1062). Providence, USA. doi:10.1145/1559845.1559981.

Brodsky, A., Egge, N., & Wang, X. (2011). Reusing relational queries for intuitive decision optimization. In *Proceedings of 44th Hawaii international conference on system sciences*. Kauai, USA. doi:10.1109/HICSS.2011.360.

Brodsky, A., Shao, G., & Riddick, F. (2014). Process analytics formalism for decision guidance in sustainable manufacturing. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0892-9.

DOC. (2014). How does commerce define sustainable manufacturing. Web page. Department of Commerce. http://www.trade.gov/competitiveness/sustainablemanufacturing/how_doc_defines_SM.asp. Accessed 9 July 2014.

EIA. (2014a). International energy outlook 2013. Web page. Energy Information Administration. http://www.eia.gov/forecasts/ieo/industrial.cfm. Accessed 8 July 2014.

EIA. (2014b). June 2014 monthly energy review. Resource document. Energy Information Administration. http://www.eia.gov/totalenergy/data/monthly/pdf/mer.pdf. Accessed 20 July 2014.

Fujitsu. (2011). Fujitsu offers energy-saving green infrastructure solution. Web page. Fujitsu. http://www.fujitsu.com/global/news/pr/archives/month/2007/20071210-02.html. Accessed 1 July 2014.

Gill, Philip E., Murray, W., & Saunders, M. (2008). User's guide for SNOPT version 7: Software for large-scale nonlinear programming. White paper. University of California San Diego. http://web.stanford.edu/group/SOL/guides/sndoc7.pdf. Accessed 9 June 2014.

GM. (2013). Innovation: Environment. Web page. General motors. http://gmsustainability.com. Accessed 2 July 2014.

Heilala, J., Saija, V., Tonteri, H., Montonen, J., Johansson, B., & Stahre, L. (2008). Simulation-based sustainable manufacturing system design. In *Proceedings of 2008 winter simulation conference* (pp. 1922–1930). Austin, USA.

IBM. (2014). CPLEX optimizer. Web page. IBM. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer. Accessed 9 July 2014.

ISO/IEC. (2011). Information technology—database languages-SQL—Part 1: Framework (SQL/Framework). Resource document. International Standards Organization and International Electrotechnical Commission. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53681. Accessed 18 July 2014.

Johansson, M., Leong, S., Lee, Y., Riddick, F., Shao, G., Johansson, B., & Klingstam, P. (2007). A test implementation of the core manufacturing simulation data specification. In *Proceedings of 2007 winter simulation conference* (pp. 1673–1681). Washington, DC, USA.

Katherasan, D., Elias, J., Sathiya, P., & Haq, A. N. (2014). Simulation and parameter optimization of flux cored arc welding using artificial neural network and particle swarm optimization algorithm. *Journal of Intelligent Manufacturing*, *25*(6), 67–76. doi:10.1007/s10845-012-0675-0.

Kim, D., Shin, S., Shao, G., & Brodsky, A. (2014). A decision guidance framework for sustainability performance analysis of manufacturing processes. NIST interagency/internal report (NISTIR)-7984.

Last, M., Danon, G., Biderman, S., & (2009). Optimizing a batch manufacturing process through interpretable data mining models. *Journal of Intelligent Manufacturing, 20*(5), 523–534.

Law, A., & Kelton, W. (2000). *Simulation modeling and analysis*. Boston: McGraw-Hill Education.

Naeem, M. A., Dias, D. J., Tibrewal, R., Chang, R. C., & Tiwari, M. K. (2013). Production planning optimization for manufacturing and remanufacturing system in stochastic environment. *Journal of Intelligent Manufacturing*, *24*(4), 717–728.

Peng, T., Xu, X., & Wang, L. (2014). A novel energy demand modeling approach for CNC machining based on function blocks. *Journal of Manufacturing Systems*, *33*(1), 196–208.

Philips. (2012). Philips sustainability statements. Web page. Philps. http://www.annualreport2012.philips.com/annual_report_2012/en/sustainability_statements.aspx. Accessed 4 July 2014.

Rachuri, S. (2010). Metrics, standards, and infrastructure for sustainable manufacturing. Presentation slides. NIST workshop on sustainable manufacturing. http://www.mel.nist.gov/msid/conferences/talks/rsudarsan.pdf. Accessed 28 July 2014.

Ridwan, F., Xu, X., & Liu, G. (2012). A framework for machining optimization based on STEP-NC. *Journal of Intelligent Manufacturing*, *23*(3), 423–441.

Rockwell Automation. (2014). Capabilities: Sustainable production. Web page. Rockwell automation. http://www.rockwellautomation.com/solutions-services/capabilities/sustainable-production/overview.page. Accessed 9 July 2014.

SAIC. (2011). SAIC to present at autovation on utility metering, monitoring, and control systems. Web page. Science Applications International Corporation. https://www.leidos.com/announcement/saic-present-autovation-utility-metering-monitoring-and-control-systems. Accessed 27 July 2014.

Shao, G., Brodsky, A., Ammann, P., & McLean, C. (2009). Parameter validation using constraint optimization for modeling and simulation. In *Proceedings of the industrial simulation conference 2009* (pp. 323–327). Loughborough, UK.

Shao, G., Bengtsson, N., & Johansson, B. (2010). Interoperability for simulation of sustainable manufacturing. In *Proceedings of 2010 spring simulation multi-conference*, Orlando, USA. doi:10.1145/1878537.1878595.

Skoog, A. (2009). *Methods for input data management: Reducing the time-consumption in discrete event simulation*. Gothenburg, Sweden: Chalmers University of Technology.

Smith, L., & Ball, P. (2012). Steps towards sustainable manufacturing through modelling material, energy and waste flows. *International Journal of Production Economics*, *140*(1), 227–238. doi:10.1016/j.ijpe.2012.01.036.

SMLC. (2011). Implementing 21st century smart manufacturing. White paper. Smart manufacturing leadership coalition. https://smart-process-manufacturing.ucla.edu/about/news/Smart%20Manufacturing%206_24_11.pdf. Accessed 19 July 2014.

Solding, P., Petku, D., & Mardan, N. (2009). Using simulation for more sustainable production systems—methodologies and case studies. *International Journal of Sustainable Engineering*, *2*(2), 111–122. doi:10.1080/19397030902960994.

Tari, H. M., & Söderström, M. (2002). Modelling of thermal energy storage in industrial energy systems the method development of MIND. *Applied Thermal Engineering*, *22*(11), 1195–1205. doi:10.1016/S1359-4311(02)00044-3.

Wang, G., Wang, Y., & Zhao, J. (2012). Process optimization of the serial-parallel hybrid polishing machine tool based on artificial neural network and genetic algorithm. *Journal of Intelligent Manufacturing*, *23*(3), 365–374.

Yang, L., Deuse, J., & Jiang, P. (2013). Multi-objective optimization of facility planning for energy intensive companies. *Journal of Intelligent Manufacturing*, *24*(6), 1095–1109.