Proceedings of the ASME 2013 International Design Engineering Technical Conference& Computers and Information in Engineering Conference
IDETC/CIE 2013
August 4-7, 2013, Portland, Oregon, USA

DETC2013-12250

SUSTAINABLE PROCESS ANALYTICS FORMALISM: A CASE STUDY OF BOOK BINDING SYSTEM FOR ENERGY OPTIMIZATION

Duck Bong Kim[1], Guodong Shao
Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD, USA
[1]Contact Author

Alexander Brodsky
Dept. of Computer Science
George Mason University
Fairfax, VA, USA

Ryan Consylman
Dept. of Computer Science
Millersville University
Millersville, PA, USA

ABSTRACT

Energy is considered as one of the important factors for manufacturers to achieve the sustainability objective. To improve energy efficiency in manufacturing, optimization techniques are essential to provide decision support. However, formulating and solving energy optimization in manufacturing is still time-consuming and difficult due to its complexity with a broad scope. In addition, it is a challenging task since it requires substantial development efforts and modeling expertise. To address this drawback, Sustainable Process Analytics Formalism (SPAF) is proposed to facilitate the modeling and optimization. In this paper, SPAF will be applied to a case study of energy optimization for a book binding production system for its feasibility validation. The knowledge of process flow, data, and metrics of the case study is represented using SPAF, and a preliminary analysis of optimization results was performed.

KEYWORDS

Sustainable Manufacturing, Sustainable Process Analytics Formalism (SPAF), Optimization, Book Binding System, Energy

INTRODUCTION

Sustainability is becoming a major concern around the globe due to the emerging issues such as depletion of material resources and nonrenewable energy concerns [1]. To address sustainability globally, many international organizations such as the United Nations (UN), European Union (EU), Organization for Economic Co-operation and Development (OECD), and researchers at the academia and government agencies have been actively researching sustainability [2]. In industry, sustainable manufacturing motivates manufacturers to consider environmental integrity, financial profitability, and social equity (e.g., energy, material, cost, and human health) [3]. Among them, energy is one of the key factors to foster the sustainability paradigm in manufacturing [4].

In sustainable manufacturing, many issues are related to solving specific optimization problems [5], which are usually complex and broad [6]. Particularly, modeling and representing them is a difficult task since it requires background knowledge about sustainable manufacturing, operational research, and methodology with a contribution of multiple experts. This results in difficulties for manufacturers, especially the small and medium enterprises (SMEs), to formulate and solve optimization problems. Thus, a comprehensive methodology that can effectively formulate and represent the optimization problems is necessary to advance sustainable manufacturing.

To address the industrial needs, Sustainable Process Analytics Formalism (SPAF) has been recently proposed by National Institute of Standards and Technology (NIST). The detail description of SPAF is out of scope in this paper, but the overview will be given. SPAF provides a mechanism to represent the knowledge of process flow, data, and mathematical specification of metrics of processes, products, and resources for sustainability analysis. It is designed to provide a unified modeling capability that enables modularity and reusability for sustainable process modeling.

The contribution of this paper is to explain a preliminary study on energy optimization for a book binding system to demonstrate feasibility of SPAF. In addition, a procedure is developed to conduct a SPAF case study, which requires neither a strong background on operation research or extensive mathematical knowledge for general users. An energy optimization problem of a book binding system is modeled into a number of modular SPAF models, and they are translated to a standard optimization model, i.e., the Optimization Programming Language (OPL) [7].

BACKGROUND AND RELATED WORK

Optimization problems are classified into the types of variables (continuous or integer) and constraints (linear or nonlinear). If the constraint is linear with continuous variables, it is considered as a linear programming (LP). In contrast, a nonlinear programming (NLP) problem studies the case in which any of the constraints or the objects function is nonlinear with continuous variables. When both continuous and integer variables are included with linear constraints, it is categorized into a mixed-integer linear programming (MILP). Similarly, a mixed-integer nonlinear programming (MINLP) studies the case when both continuous and integer variables are included and the constraint is nonlinear. Thus, different types of optimization problems require different types of optimization tools and mathematical models via a user-friendly interface [8].

The optimization tools for sustainable manufacturing can be categorized into a numerical-based optimization tool, a simulation-based optimization tool, and a graph-based optimization tool [9]. A numerical-based tool consists of a solver and a modeling environment. A solver is a component of optimization software that solves mathematical problems using solution methods. On the other hand, a modeling environment provides general and intuitive ways to express mathematical problems, generates problem instances, and interfaces to other applications. A simulation-based optimization tool is a software package that integrates optimization techniques with simulation software, which consists of an optimization package and its simulation platform. A graph-based optimization tool is suitable for representing and analyzing a complex system by reducing some possibilities of candidate operating units.

To model a complex system for analysis, a modeling language is required. Among many process modeling languages [10], Process Specification Language (PSL) [11], Business Process Modeling Notation (BPMN) [12], Systems Modeling Language (SysML) [13], and Object Process Methodology (OPM) [14] are selected and its relevance is explained. PSL is a neutral language designed to support exchange information among various manufacturing applications (e.g., process modeling, process planning, and simulation). SysML is a general purpose modeling language that helps to analyze, design, verify, and validate a wide spectrum of engineering applications. BPMN offers a notation for process description that includes the visual appearance and the semantics of the elements. OPM is an integrated approach that enables one to design complex systems (e.g., systems architecting) using both graphic and textual language.

Recently, there are many global research efforts to develop a framework and database that can assess and reduce the environmental burden such as CO2PE!-Initiative (Cooperative Effort on Process Emissions in Manufacturing) [15-17] and Ecoinvent [18]. The CO2PE!-Initiative is intended to promote a coordinated international effort and collect, document, and analyze the environmental impacts for a range of manufacturing processes. Guidelines are provided to reduce environmental impacts for conventional and emerging manufacturing processes. The ecoinvent database contains lifecycle inventory (LCI) data [18], widely used in LCI data exchange format and popularly supported by LCA software systems. However, there is no comprehensive methodology or global research efforts that help solve optimization problems or make decisions for improvements in sustainable manufacturing (e.g., energy and material efficiency, waste and emission control, and cost reduction) from the optimization perspective.

OVERVIEW OF SUSTAINABLE PROCESS ANALYTICS FORMALISM

In this section, the requirements and model components of SPAF will be briefly explained.

Requirements

Manufacturing processes and systems become more and more complicated due to the expanded activities and the dynamics of manufacturing environment. This means that analysis performance such as optimization and decision support problems for sustainable manufacturing usually have a wide spectrum of mathematical structures. In addition, optimization or decision support problems usually have characteristics such as a multi-dimensional optimization problem or computational-intensive mathematical operations [19]. For example, many optimization algorithms for sustainable manufacturing are to minimize or maximize the environmental, economic, and societal factors simultaneously. This also increases the complexity for manufacturers to formulate and solve analysis problems.

Figure 1 shows the main motivation of SPAF by comparing the current modeling approaches with SPAF. Many current approaches perform the analysis as shown in Figure 1 (left). In other words, they model the data with a specific tool or environment. This results in difficulties and requires substantial efforts to perform the analysis for the same problem. In contrast, SPAF provides a unified modeling environment for different types of data sets (e.g., material information, energy consumption, and process plan) and the various analysis applications as shown in Figure 1 (right). It is formalism for process language and mathematical modeling, which enables users to transform the collected sustainability data into the computation-friendly structure for analysis.

To deal with the complexity in sustainable manufacturing processes, partitioning the problem along a logical boundary into a collection of components or modules is a good way since it can increase understanding and reusability as a part of other problems. Thus, modularity and reusability features for SPAF models can increase the effectiveness of formulation and performance of problem analysis; hence a significant amount of time and efforts on model development and analysis could be saved.

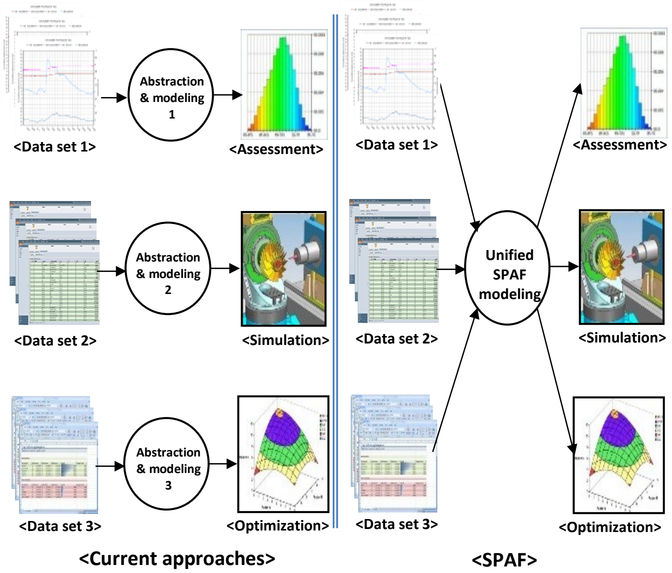

Figure 1. MOTIVATION OF SPAF.

SPAF Model Components

SPAF consists of four model components including context, flow, flow aggregator, and process as shown in Figure 2.

Context is composed of overall information (e.g., context ID and global variable), which can be used by other SPAF models. Flow is related to the inputs and outputs of processes. In the case of a milling machine [20], the work piece can be as an input flow of material, and the product and the metal chip can be as an output flow. Flow aggregator is used to aggregate the same types of outputs from processes as input and distribute the outputs as inputs of other processes. For example, energy consumption or waste from multiple processes can be aggregated into a total energy consumption or waste. A process can be considered as a main process and its sub-processes. For one manufacturing process, there can be multiple sub-processes within it.

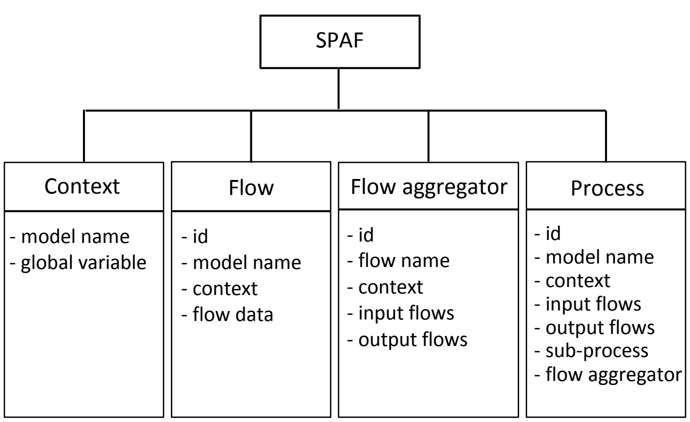

Figure 2. MODEL COMPONENTS OF SPAF.

OVERALL PROCEDURE

To conduct a case study on an optimization for a book binding system, a procedure is developed.

Figure 3 shows the overall flow of the procedure, which consists of four steps: problem formulation, data collection, modeling and optimization performance, and actionable recommendations. The procedure starts with problem formulation. Scoping is necessary to define and simplify the boundary of an optimization problem. The constraints and objectives are the information for modeling the mathematical abstraction to build an objective function. Data collection is an important procedure since the accuracy of the collected data can significantly affect the results of an optimization problem. Data is collected with respect to the selected sustainable manufacturing indicators and metrics. The modeling and optimization performance procedure formulates and represents a given problem, and performs decision optimization. It consists of the Sustainable Process Analytics Formalism (SPAF), translator, optimization modeling environment, and optimization solver. After modeling and optimization performance, actionable recommendations are provided in the final stage.

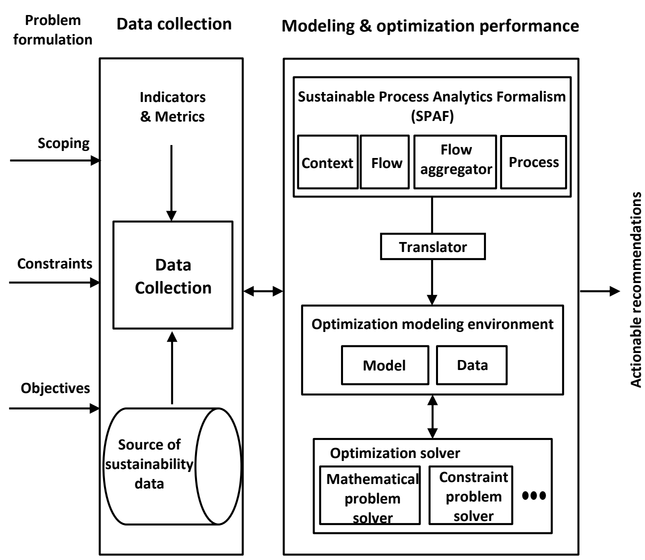

Figure 3. THE OVERALL PROCEDURE OF OPTIMIZATION PERFORMANCE USING SPAF.

Data Collection

It is a time-consuming procedure to collect sustainability data. From the sustainable manufacturing perspective, indicators and metrics for a given task are selected to perform optimization. With respect to the different indicators and metrics, the objective functions are different and lead to different optimal results. Determination of weight factor and normalization for each indicator also affects the optimal results [21]. Data collection from the source of sustainability data can be categorized into the quantitative and qualitative sustainability data [22]. For example, the energy consumption, labor cost, energy cost, and water usage are quantitative while the human health, operational safety, and ergonomics are the qualitative.

There are several ways to acquire the quantitative sustainability data, for example:

- Onsite when manufacturers have the data.
- Measurement devices directly (e.g., MTConnect, OPC, and Online monitoring/logging of PLC signals) [23].
- Simulation applications. For example, a discrete event simulation (DES) tool can provide the sustainability data without real measurements.
- Empirical prediction when some variables are unknown.
- Analytical calculation when knowing the mathematical model.
- Repository of sustainability data such as CO2PE! [15] or Ecoinvent [18].

For the qualitative sustainability data, there are two approaches. The data can be obtained based on the soft computing techniques such as fuzzy-logic, neural network, and genetic algorithm [24]. The soft computing techniques provide the capability to convert subjective knowledge/opinion into a mathematic formulation. Second, it can be obtained from the repository of sustainability data.

Modeling and optimization performance

The book binding system can be described and formalized by using a process modeling language (e.g., PSL [11], BPMN [12], SysML [13], and OPM [14]). However, to validate feasibility, SPAF is used to represent the flow, data, process, and metrics in this case study. The flow, data, and processes of the book binding system are described using three model components: flow, flow aggregator, and process. Specific examples of the SPAF models are shown in the following section. In addition, a well-defined syntax sample codes for each model are shown in the Appendix.

To perform analysis from the SPAF models, an interface tool to translate the SPAF models into a specific analysis application is required. In this paper, the translator transforms the SPAF model into a standard optimization model structure in Optimization Programming Language (OPL) [7].

The optimization modeling environment mediates between users and optimization solvers. It provides general and intuitive ways to express mathematical problems, generates instances, offers features for importing data, invokes solvers, analyzes results, and interfaces to other applications. The OPL is a modeling language designed to optimize a given problem. The modeling environment consists of the model file and data file. The mathematical models (e.g., constraint and objectives) are the inputs for the model file while the collected data are the inputs for the data file.

The solver performs optimization problems such as mathematical (MP) or constraint problems (CP) using solution methods and returns the optimal results. A number of optimization solvers and modeling environment packages have already been developed as separate applications. Optimization solvers commonly are designed to link to different modeling environments, and modeling environments also support the use of many different types of optimization solvers.

CASE STUDY OF A BOOK BINDING SYSTEM

In the U.S., phone book manufacturers produce over 500 million phone books a year, which are produced at a cost of 19 million trees, 0.73 billion *kg* of paper, 205 m^3 of landfill, and 3.2 billion *kWh* of electricity [25]. Although there are many sustainable factors within a book binding system (e.g., paper wastes, glue, emissions, production, and operational health), this case study focuses on energy consumption due to its importance from the sustainable manufacturing perspective.

Figure 4 shows the overall process flow of a book binding system for phone book production. Manufacturing begins with the gathering process in which sections of books are drawn from hoppers and the gathered books are delivered to a binding process. In the binding process, there are two sub-processes: glue application and cover binding. Within the two sub-processes, the gathered book receives a cover and they are bound together. A stacking process stacks the books to a certain height before ejecting them to a trimmer. Within the trimming process, which is the most maintenance intensive, a stack of books is trimmed to the required size. During the wrapping process, the trimmed stacks are bound with heat-shrink plastic. In the final palletizing stage, the wrapped books are arranged onto pallets for shipping.

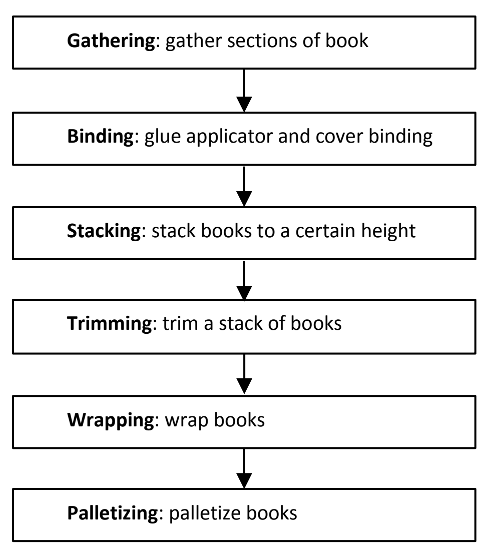

Figure 4. THE PROCESS FLOW OF A BOOK BINDING SYSTEM.

Problem Formulation

In a book binding system, energy consumption is dependent on quantity demanded, page count, and line speed. While the quantity demanded and page count are given by customers, the line speed is a primary control factor that can be used to analyze and predict energy consumption. The line speed is related to the processing time, labor cost, and waste allocation. In this case study, the line speed is only considered as the important factor for energy consumption for its simplicity. The goal of this case study is to find the optimal line speed to minimize energy consumption with respect to the quantity demanded and page count. Thus, the line speed is the factor, which is expressed as a decision variable.

Energy consumption of a certain manufacturing process is characterized based on the time study in Kellens et al. [16, 17]. The manufacturing process consists of four main modes: setup, idle, active, and teardown modes. The setup mode is for preparing the processing (e.g., cleaning and loading books). The idle mode is required to warm up the book binding system for making books (e.g., machine adjustments and clearing jams). During the active mode, papers are gathered, bound, stacked, trimmed, wrapped as well as palletized, and books are produced; while the teardown stage is to finish the process. Therefore, the total energy consumption for operating the book binding system consists of energy consumption from setup, idle, active, and teardown modes. It is assumed that energy consumption during the setup, idle, and teardown modes are constant whereas that of active mode is the variable in this case study. Unavailable data is obtained from workers' experiences and assumptions.

The book binding system consists of six sub-processes and each sub-process has four modes. Thus, the total energy (E_{total}) for the whole book binding system is calculated by

$$E_{total} = \sum_{i=1}^{6}\sum_{j=1}^{4} E_{i,j} \tag{1}$$

The total energy consumption for each sub-process ($E_{sp,total}$) is calculated using the four operating modes as below:

$$E_{sp,total} = E_{sp,setup} + E_{sp,idle} + E_{sp,active} + E_{sp,teardown} \tag{2}$$

From Equation (2), the energy consumption of each sub-process ($E_{sp,\ total}$) during setup ($E_{sp,\ setup}$), idle ($E_{sp,\ idle}$), and teardown ($E_{sp,\ teardown}$) modes are constant and will be given, while that of active mode ($E_{sp,\ active}$) is determined by

$$E_{sp,active} = P_{sp,active} \times t_{sp,active} \tag{3}$$

Power and time for active mode are estimated to formulate this optimization problem. It is assumed that the book binding system is driven by two 373 kW motors. From reference [26], the active power of the sub-process ($P_{sp,\ active}$) can be estimated in proportion to the different line speeds (unit/h). The active power for each sub-process ($P_{sp,\ active}$) is calculated by

$$P_{sp,active} = 0.00173 \times Volts \times Amps \times Power\ factor \tag{4}$$

For the calculation of active power consumption, a Siemens 373 kW electric motor is used. Its voltage and ampere are 4160 V and 60 A, respectively. The power factor is defined as the ratio of the active power (*Watts*) to the apparent power (*Volts*×*Amps*). Table 1 shows the power consumption with respect to the different line speeds (*unit/h*).

Table 1. POWER CONSUMPTION WITH RESPECT TO DIFFERENT LINE SPEEDS.

Speed (*unit/h*)	% Amps	Power Factor	Power (*kW*)
6000	100	0.95	410.21
5500	91	0.94	369.37
5000	83	0.93	333.31
4500	74	0.92	293.97
4000	66	0.90	256.49
3500	58	0.85	212.88
3000	50	0.78	168.41
2500	41	0.55	97.37
2000	33	0.30	42.75

The active time ($t_{sp,\ active}$) for a sub-process is calculated by

$$t_{sp,active} = \frac{demandQty}{lineSpeed} \tag{5}$$

The objective function is to minimize the total energy consumption for the book binding system while meeting quantity demanded. It is a function of three independent variables including line speed "*lineSpeed*" (units/h), quantity demanded, and active power consumption. The line speed is decision variable while the "*demandQty*" is a given variable from a customer. The "$P_{sp,active}$" is independent variable, but it is determined with respect to the line speed. The objective function and constraint are shown as below:

$$min(E_{total}) = f(lineSpeed, demandQty, P_{sp,active}) \tag{6}$$

$$2000.0 \leq lineSpeed \leq 6000.0 \tag{7}$$

SPAF Modeling

From the general procedure of the book binding system, Figure 5 shows the schematic diagram of input, output, flow, and process of the book binding system. It starts with the printed paper (input) and ends with the final product (phone book). Each process receives the pre-processed paper and consumes the energy while it generates wastes and produces output. The total energy consumption (E_{total}) is the summation of the six sub-processes; total waste (W_{total}) is the summation of the wastes from the six sub-processes. The book binding system can be modeled by using flow, flow aggregator, and process description components in SPAF models (see Appendix). Although waste is an important factor of a book binding system, optimization for waste is not performed in this case study.

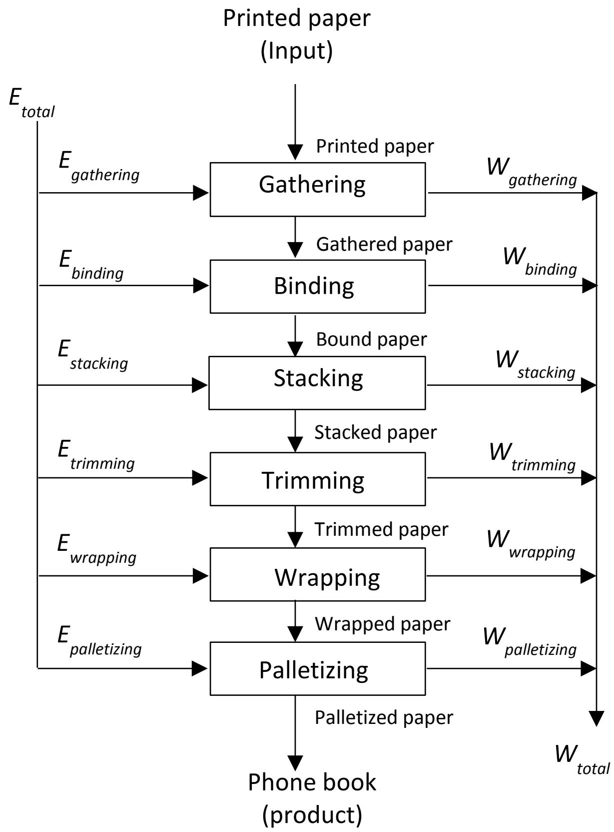

Figure 5. A SCHEMATIC DIAGRAM OF INPUT, OUTPUT, FLOW, AND PROCESS OF THE BOOK BINDING SYSTEM.

The resource flows "paper", "energy", and "waste" can be modeled by using a flow model. The paper is modeled using "flow item(*Id*)" while energy and waste can be modeled by continuous flow "flow contFlow(*Id*)" as shown in Figure A in the Appendix.

The total consumption of energy, paper, and waste are modeled by using a flow aggregator model. The input flow and output flow of a piece of paper is modeled and constraint is given in "flow aggregator itemAggr(*Id*)" while those of energy and waste are modeled and constraints are given in "flow aggregator contFlowAggr(*Id*)" as shown in Figure B in the Appendix.

Since the sub-processes including gathering, binding, stacking, trimming, wrapping, and palletizing have the same features, a generic model for each sub-process is generated by using a process model. Each sub-process utilizes the generic model and is also modeled by a process model. Figure 6 shows the schematic diagram of a generic process model for each of sub-processes. It shows the energy consumption for sub-process ($E_{sp,total}$), pre-processed item, processed item, and waste during the process ($W_{process}$). The energy consumption for the sub-process includes four stages: setup, idle, active, and teardown. The generic sub-process can be represented by the SPAF "process" model. The sample syntax codes are shown and explained in Figure C in the Appendix.

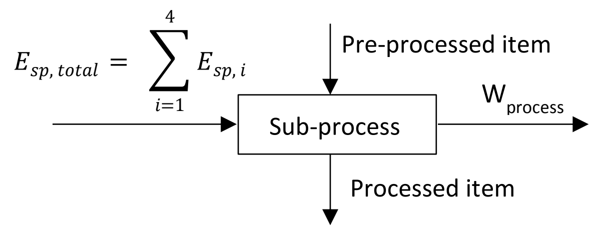

Figure 6. A SCHEMATIC DIAGRAM OF A GENERIC SUB-PROCESS.

Figure 7 shows a schematic diagram to explain the details of the sub-process "*gathering*". This gathering process utilizes the generic sub-process model using "include generic_unit(*Id*)". The gathering process receives the printed paper and requires the energy ($E_{gath.,total}$) during setup, idle, active, and teardown modes. It also generates waste during the sub-process and outputs the gathered paper. The gathering process is represented by using the SPAF process model. The sample syntax code for gathering is shown and explained in Figure D in the Appendix.

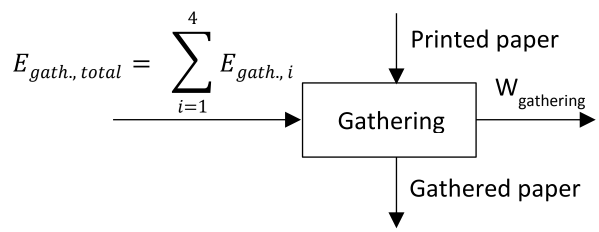

Figure 7. A SCHEMATIC DIAGRAM OF THE SUB-PROCESS "GATHERING".

The whole book binding system can be modeled by using a main process component, expressed as "process binding_line(*Id*)". In the model, input and output flows of six sub-processes, their aggregators, and their mathematical formula for total energy consumption are included. The constraint is also represented in the model. The optimization query is shown with "binding_line(*Id*)" and "min totalEnergy". The "min totalEnergy" denotes the objective function to find the optimal line speed to minimize the energy consumption for the whole book binding system. The sample syntax codes are shown in Figure E in the Appendix.

Expression in OPL

The SPAF models are manually transformed to OPL model and data files. The transformed OPL is solved by using IBM ILOG CPLEX Optimization Studio [27]. Figure F and Figure G in the Appendix show some part of the transformed code of model and data files in an OPL environment, respectively.

Table 2 shows the results of optimal line speed (units/h) and its energy consumption (E_{total}) with respect to the quantity demanded and page count. It is shown that the optimal line speed is around 4000 units/h. Figure 8 shows the relationship between the energy consumption and quantity demanded. It is shown that the energy consumption is proportional to the quantity demanded. The active energy linearly determines the total energy consumption with respect to the quantity demanded while the setup, idle, and teardown energy are the same in all cases.

Table 2. RESULTS OF OPTIMAL SPEED WITH RESPECT TO THE DEMAND QUANTITY AND PAGE COUNT.

Quantity demanded	Page count	Energy consumption (kWh)	Optimal line speed (units/h)
1 000	200	1841.1	4004
5 000	200	2005.4	4004
10 000	200	2210.8	3999
30 000	200	3032.5	4004
50 000	200	3853.8	4000
100 000	200	5907.8	3999
300 000	200	14123.4	3999
500 000	200	22339.0	3999
1 000 000	200	42878.0	3999

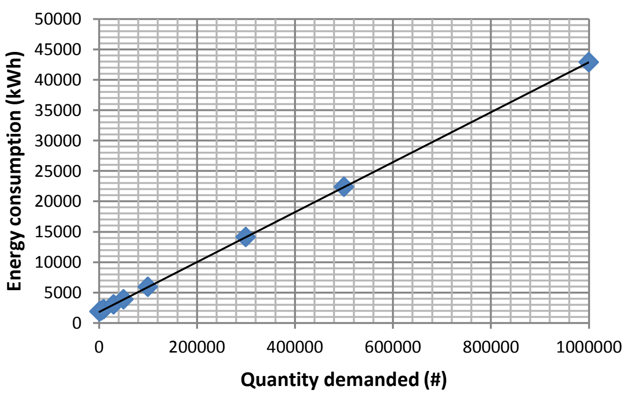

Figure 8. RELATIONSHIP BETWEEN QUANITY DEMANDED AND ENERGY CONSUMPTION.

DISCUSSION & CONCLUSION

In this paper, a preliminary case study on energy optimization for a book binding system is conducted to validate the feasibility and usability of SPAF. A procedure is also presented, which consists of four stages: problem formulation, data collection, modeling and optimization performance, and actionable recommendations. An energy optimization problem of a book binding system is modeled into a number of modular SPAF models such as flow, flow aggregator, and process model components. The SPAF models are manually translated into an OPL model. It is shown that the optimal line speed is around 4000 units/h.

As applied to an energy optimization problem for a book binding system, SPAF can represent, model, and describe sustainable manufacturing processes, flow, data, and metrics for other sustainable manufacturing processes. In addition, it is intentionally designed to have features including reusability and modularity. Based on SPAF, sustainable manufacturing processes can be modeled in a modular way by using context, flow, flow aggregator, and process models. This can enhance reusability when some models or problems are similar or the same. These features can increase usability of SPAF for analysis purposes to advance sustainable manufacturing.

In the near future, a translator will be developed, which can automatically transform the SPAF models into a standard optimization model structure or format for its optimization analysis. In addition, the case study will be expanded to other sustainability indicators and metrics such as material consumption, wastes, and air emission.

DISCLAIMER

Certain company names or commercial products may have been identified in this paper. Such identification was used only for illustration purposes. This use does not imply approval nor endorsement by NIST. Furthermore, it does not imply that such company names and products are necessarily the best for the purpose.

REFERENCES

[1] World Commission on Environment and Development (WCED) (1987). Our Common Future. Oxford University Press, Oxford, UK, New York, NY, USA.
[2] Singh, R.K., H.R. Murty, S.K. Gupta, and A.K. Dikshit (2012). "An overview of sustainability assessment methodologies." Ecological Indicators 15(1), pp. 281-299.
[3] Sustainable Manufacturing Program at National Institute of Standards and Technology. http://www.nist.gov/el/msid/lifecycle/sustainable_mfg.cfm
[4] Rachuri, S., R.D. Sriram, A. Narayanan, P. Sarkar, J.H. Lee, K.W. Lyons, V. Srinivasan, and S.J. Kemmerer (2011). "Summary of the NIST workshop on sustainable manufacturing: metrics, standards, and infrastructure." International Journal of Sustainable Manufacturing, 2(2/3), pp. 237-259.
[5] Jawahir, I.S. and O.W. Dillon (2007). "Sustainable manufacturing processes: new challenges for developing predictive models and optimization techniques." Proceedings of 1st international conference on sustainable manufacturing SM1, Montreal, Canada, October 17-18.
[6] Bi, Z. (2011). "Revisiting System Paradigms from the Viewpoint of Manufacturing Sustainability." Sustainability, 3(9), pp. 1323-1340.
[7] Hentenryck, P.V. (1999). "The OPL optimization programming language." January 8, 1999, the MIT Press.
[8] NEOS, http://neos-guide.org/optimization-tree

[9] Klemes, J. (2011). "Sustainability in the Process Industry: Integration and Optimization." The McGraw-Hill Companies, Inc.
[10] Anderl, R., J. Malzacher, and J. Raßler (2008). "Proposal for an Object Oriented Process Modeling Language." Enterprise Interoperability III, Part VI, pp. 533-545.
[11] International Standards Organization (ISO): ISO 18629 Series: Process Specification Language of 2004, www.iso.org.
[12] White, S.A. and D. Miers (2008). "BPMN Modeling and Reference Guide: Understanding and Using BPMN." Future Strategies Inc.
[13] Friedenthal, S., A. Moore, and R. Steiner (2008). "A practical guide to SysML: The systems modeling language." Morgan Kaufmann (August 7, 2008), Elsevier Inc.
[14] Dori, D. (2002). "Object-Process Methodology - A Holistic Systems Paradigm." Springer Verlag, New York.
[15] CO2PE! Cooperative effort on process emissions in manufacturing website. http://www.mech.kuleuven.be/co2pe.
[16] Kellens, K, W. Dewulf, M. Overcash, M. Hauschild, and J.R. Duflou (2011). "Methodology for systematic analysis and improvement of manufacturing unit process life cycle inventory (UPLCI) Part 1: Methodology Description." International Journal of Life Cycle Assessment 17(1), pp. 69-78.
[17] Kellens, K, W. Dewulf, M. Overcash, M. Hauschild, and J.R. Duflou (2012). "Methodology for systematic analysis and improvement of manufacturing unit process life cycle inventory (UPLCI). Part 2: case studies." International Journal of Life Cycle Assessment 17(2), pp. 242-251.
[18] Ecoinvent (2011). Ecoinvent Centre, Swiss Centre for Life Cycle Inventories. Available from http://www.ecoinvent.ch
[19] Sinclair, P. (2011) "Describing the elephant: a framework for supporting sustainable development processes." Renewable and sustainable energy reviews 15, pp. 2990-2998.
[20] Narita, H. and Fujimoto, H., 2009. "Analysis of Environmental Impact Due to Machine Tool Operation." International Journal of Automation Technology, 3(1), pp. 49-55.
[21] Bohringer, C. and P.E.P. Jochem (2007). "Measuring the immeasurable - a survey of sustainability indices." Ecological Economics 63(1), pp. 1-8.
[22] Lu, T., A. Gupta, A.D. Jayal, F. Badurdeen, S.C. Feng, O.W. Dillon, and I.S. Jawahir (2010). "A Framework of Product and Process Metrics for Sustainable Manufacturing." Proceedings of the Eighth International Conference on Sustainable Manufacturing, November 22-24, Abu Dhabi.
[23] Vijayaraghavan, A., W. Sobel, A. Fox, D. Dornfeld, and P. Warn-dorf (2008). "Improving Machine Tool Interoperability using Standardized Interface Protocols: MTConnect." International Symposium on Flexible Automation.
[24] Chandrasekaran, M., M. Muralidhar, C.M. Krishna, and U.S. Dixit (2010). "Application of soft computing techniques in machining performance prediction and optimization: a literature review." The International Journal of Advanced Manufacturing Technology 46(5-8), pp. 445-464.
[25] Metro (2012), "Too many phone books?" http://www.oregonmetro.gov/index.cfm/go/by.web/id=31588
[26] SmartEnergyUser (1996), "Calculating what it costs to run a motor." http://www.wisdompage.com/SEUhtmDOCS/SEU17.htm.
[27] IBM ILOG CPLEX Optimization Studio, http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

APPENDIX: SAMPLE SYNTAX CODES OF SPAF

Figure A shows the flow model with "*item(Id)*" and "*contFlow(Id)*". The "*item(Id)*" denotes the inputs (pre-processed) and outputs (processed) paper of a process while the "*contFlow(Id)*" denotes the continuous inputs (energy consumption) and outputs (wastes) of a process, respectively. The "*Id*" is a parameter that can be replaced via an include statement. The "*item(Id)*" declares the quantity *(Id.qty)* and page count *(Id.pageCount)* of the book while the "*contFlow(Id)*" declares the amount (*Id.amount*) of input and output of a process. The three dots "…" denotes the missing data and should be instantiated using a constant before use.

```
flow item(Id) {
int Id.qty;
int Id.pageCount = ...;
}

flow contFlow(Id) {
float Id.amount;
}
```

Figure A. FLOW MODEL COMPOENTS.

Figure B shows the flow aggregator model with "*itemAggr(Id)*" and "*contFlowAggr(Id)*", respectively. In "*itemAggr*", "*Id.itemType*" is declared as a string "*item*". "*Id.inputFlows*" and "*Id.outputFlows*" are declared as a set of string and will be instantiated, respectively. A constraint is shown that the total of inputs "*inputFlows*" is the same to the total of outputs "*outputFlows*". In "*contFlowAggr(Id)*", "*contFlow*" is declared as a string "*contFlow*". *Id.inputFlows* and *Id.outputFlows* are declared as a set of string and will be instantiated, respectively. A constraint is shown that the total of inputs "*inputFlows*" is the same to the total of outputs "*outputFlows*".

```
flow aggregator itemAggr (Id){
String Id.itemType = "item";
{string} Id.inputFlows = ...;
{string} Id.outputFlows = ...;
sum (i in Id.inputFlows) i.qty
            == sum (o in Id.outputFlows) o.qty;
}
flow aggregator contFlowAggr (Id){
String Id.itemType = "contFlow";
{string} Id.inputFlows = ...;
{string} Id.outputFlows = ...;
sum (i in Id.inputFlows) i.amount
            == sum (o in Id.outputFlows) o.amount;
}
```

Figure B. FLOW AGGREGATOR MODEL COMPONENTS.

Figure C shows the process model in the case of the generic process in SPAF. The inputs and outputs of a process are declared as a set of strings "*Id.itemIn*", "*Id.power*", "*Id.itemOut*", and "*Id.waste*". The pre-processed material is denoted as "*Id.itemIn*" while the processed item is denoted as "*Id.itemOut*". The power consumption is declared as "*Id.power*" while the waste is declared as "*Id.waste*". "*Id.lineSpeed*" is the decision variable. Then the flow models such as "*item(Id.itemIn)*", "*contFlow(Id.power)*", "*item(Id.itemOut)*", and "*contFlow(Id.waste)*" are included. The "*Id.powerFunciton*" is declared as "*pwlFunction*", which denotes the piecewise linear function to compute the power consumption for the active mode. The energy consumption for each sub-process (*Id.Energy*) is the sum of four modes: *Id.Esetup*, *Id.Eidle*, *Id.Eactive*, and *Id.Eteardown*.

```
process generic_unit(Id) {
string Id.itemIn = ...;
string Id.power = ...;
string Id.itemOut = ...;
string Id.waste = ...;
float Id.lineSpeed;
include item(Id.itemIn);
include contFlow(Id.power);
include item(Id.itemOut);
include contFlow(Id.waste);
{string} inputFlows = {Id.itemIn,Id.power};
{string} outputFlows = {Id.itemOut,Id.waste};
float Id.Esetup = ...;
float Id.Eteardown = ...;
float Id.Eidle = ...;
pwlFunction Id.powerFunction = ...;
int Id.itemIn.qty = Id.itemOut.qty;
float Id.EperPage = Id.powerFunction(Id.lineSpeed);
float Id.Eprocess
  = Id.EperPage*Id.itemOut.qty/Id.lineSpeed
float Id.Energy
  = Id.Esetup + Id.Eidle + Id.Eprocess + Id.Eteardown;
float Id.power.amount = Id.Energy;
float Id.waste.amount = 0.0;
}
```

Figure C. THE GENERIC PROCESS MODEL IN SPAF.

Figure D shows the process model for the sub-process "gathering". The string "Id.name" is declared as "gathering". The input (*Id.itemIn*) and output (*Id.itemOut*) are denoted as "printed" and "gathered", respectively. The string "*Id.power*" and "*Id.waste*" are declared as "*powerToGathering*" and "*wasteFromGathering*", respectively. The energy consumption for setup (*Id.Esetup)*, idle (*Id.Eidle*), and teardown (Id.*Eteardown*) modes are given, respectively. The "*pwlFunction*" denotes the piecewise linear function which computes the power consumption for the active mode with respect to the different line speed, as estimated in Table 1. It ends with including the "*generic_unit(Id)*" as declared in Figure C. The other sub-processes including binding, stacking, trimming, wrapping, and palletizing have the same structure as formalized in Figure D.

```
process gathering(Id) {
string Id.name = "gathering";
string Id.itemIn = "printed";
string Id.itemOut = "gathered";
string Id.power = "powerToGathering";
string Id.waste = "wasteFromGathering";
float Id.Esetup = 15.0 ;
float Id.Eteardown =21.0;
float Id.Eidle = 35.0;
pwlFunction Id.powerFunction = piecewise{
   0.0345 -> 2500; 0.1421 -> 3000; 0.0889 -> 3500;
   0.0872 -> 4000; 0.0750 -> 4500; 0.0787 -> 5000;
   0.0721 -> 5500; 0.0817} (2000, 42.75);
include generic_unit(Id);
}
```

Figure D. AN EXAMPLE OF A PROCESS MODEL OF THE SUB-PROCESS "GATHERING".

Figure E shows the main process model of the whole book binding system. The "*demantQty*" and "*pageCount*" are declared as an integer and will be instantiated, respectively. The "*lineSpeed*", a decision variable, is declared as a float number and constrained from (2000 to 6000) units/h. The "*subProcesses*" is denoted by a set of six strings such as the "*gathering*", "*binding*", "*stacking*", "*trimming*", "*wrapping*", and "*palletizing*". The "*AllBindingItems*", input flows of each sub-process, are also a set of strings. The final output from the process (*palletized.qty*) should be same to the demand quantity. The total energy (*totalEnergy*) is the sum of the six sub-processes for the book binding system. The total waste (*totalWaste*) can be computed by the sum of the wastes from the six sub-processes.

```
process binding_line(Id) {
int demandQty = ...;
int pageCount = ...;
float lineSpeed;
2000.0 <= lineSpeed <= 6000.0;
{string} inputFlows = {"totalEnergy","printed"};
{string} outputFlows = {"palletized","totalWaste"};
{string} subProcesses =
        {"gathering","binding","stacking",
        "trimming","wrapping","palletizing"};
for (s in subProcesses) {
        float s.lineSpeed = lineSpeed;
        include s();
        };
{string} AllBindingItems =
        {i | s in subProcesses, i in s.inputFlows}
union {"palletized"};
for (i in AllBindingItems) i.pageCount = pageCount;
int palletized.qty = demandQty;
{string} energyAggr.inputFlows = {"totalEnergy"};
{string} energyAggr.outputFlows = {
        "powerToGathering", "powerToBinding",
        "powerToStacking", "powerToTrimming",
        "powerToWrapping", "powerToPalletizing" };

include contFlowAggr("energyAggr");
{string} wasteAggr.inputFlows = {
        "wasteFromGathering", "wasteFromBinding",
        "wasteFromStacking", "wasteFromTrimming",
        "wasteFromWrapping", "wasteFromPalletizing" ;

{string} wasteAggr.outputFlows = {"totalWaste"};
include contFlowAggr("wasteAggr");
float totalEnergy.amount = sum (s in subProcesses)
s.Energy;
float totalWaste.amount = sum (s in subProcesses)
}
//Optimization query
include binding_line();
min totalEnergy;
```

Figure E. A PROCESS MODEL OF THE BOOK BINDING SYSTEM.

Figure F and Figure G show the sample syntax codes of OPL model and data from the SPAF model, respectively. This optimization model consists of decision variables and constraints to find a solution using a constraint programming optimization solver "*using CP*". The data structures can be constructed using "*tuple*" that clusters related data. The decision variable, expressed as "*dvar*", satisfies all constraints and optimize a specific objective function. The decision expression, expressed as "*dexpr*", is used to have a meaning with respect to the original problem, writing it as a decision expression makes the model more readable. The "*minimize*" is to express the objective function.

```
using CP;
tuple OrderData{
   string  OrderName;
   int     demandQty;
   int      pageCount;
}
{OrderData} Orders = ...;
int MinSpeed = ...;
int MaxSpeed =...;
dvar int lineSpeed[Orders] in MinSpeed .. MaxSpeed;

//gathering
float EsetupGathering = ...;
float EidleGathering = ...;
float EteardownGathering = ...;
dexpr float powerFunctionGathering [o in Orders] = piecewise{
        0.0345 -> 2500; 0.1421 -> 3000; 0.0889 -> 3500;
        0.0872 -> 4000; 0.0750 -> 4500; 0.0787 -> 5000;
        0.0721 -> 5500; 0.0817} (2000, 42.75) lineSpeed[o];
dexpr float EactiveGathering [o in Orders] =
      powerFunctionGathering[o] * o.demandQty /lineSpeed[o] ;
dexpr float EnergyGathering [o in Orders] =
            EsetupGathering + EidleGathering +
            EactiveGathering[o] + EteardownGathering;
                        •
                        •
///total energy            •
dexpr float totalEnergy [o in Orders] =
   EnergyGathering[o]+EnergyBinding[o]+EnergyStacking[o]+
   EnergyTrimming[o]+EnergyWrapping[o]+EnergyPalletizing[o
];
minimize
   sum(o in Orders)
      totalEnergy[o];
```

Figure F. THE TRANSFORMED CODE IN OPL MODEL FILE.

```
Orders = {
        < "NIST", 10000, 200 >
        };

MinSpeed = 2000;
MaxSpeed = 6000;

EsetupGathering = 15.0;
EidleGathering = 21.0;
EteardownGathering = 35.0;
                •
                •
                •
```

Figure G. THE TRANSFORMED CODE IN OPL DATA FILE.