

# Cognitive Work in Future Manufacturing Systems: Human-centered AI for Joint Work with Models

Peter Denno, NIST

**Abstract**—Manufacturers perpetually adapt their systems to meet unforeseen events, new objectives, competition, and improved understanding of processes. In that human-directed work, models mediate an enduring relationship between production resources and engineers. Accommodating new understanding in the models controlling production can lead to more effective manufacturing. That work has previously been the province of quality programs such as Six Sigma, but is now fertile ground to study human-computer interaction about that enduring relationship mediated by models. Can AI augment human capability in the arcane work of formulating and refining models? This question is relevant to complex system engineering generally, not just manufacturing. In answering this question, this paper adapts Klein’s *flexecution* for use in adaptable manufacturing systems. *Theory flexecution*, the methodical refinement of models, points to human-computer interactions that emphasize the roles of models, explanation, and machine agents that recognize the engineer’s goals. This perspective article illustrates these ideas with an example of formulating models for production scheduling.

## I. INTRODUCTION

To improve system performance, engineers continually adapt manufacturing systems while the systems are in service; this has been true of manufacturing for several decades [1]. The performance of other system types might be improved were they also adaptable by skilled users. One can imagine, for example, the value of a surgeon perfecting the operation of a robotic-assisted surgery system. But what characteristics of the system would enable this? Increasingly, the answer to this question is the ability for joint (AI/human) work with models of the system. Broadly speaking, today automation drives much of production, and models drive both automation and the routine manual activities of production. Today, the fitness of system models in the face of impending change is a principal element of manufacturing and industrial engineers’ situation awareness.

The notion of model intended here is a broad one, encompassing “actionable” mental models, programs, and (sometimes executable) formal models of decision processes, plans, and digital twins. Models, whether they are mental or formal, serve to routinize production; models are what make automation tick.

The act of conceiving models, variously known as *modeling* or *formulation*, is a hallmark of intelligent behavior and situation awareness [2]. It is a perennial topic of the philosophy of science. One may argue that machine agents are capable of conceiving of models; for example, unsupervised learning and plan recognition [3] are abstractive processes essential to formulation. But, of course, these processes are not self-acting.

Models can be shared and thereby help stakeholders establish common ground about the subject the models concern.

Models can serve the role of *public object* in constructionist learning [4]–[6]. This perspective article asks whether predictable patterns and conventions used in modeling in specific settings can be recognized by AI agents and thereby the agents assist in modeling tasks. Specifically, is there a role for machine agents in the tasks of formulating and refining the models driving production processes?

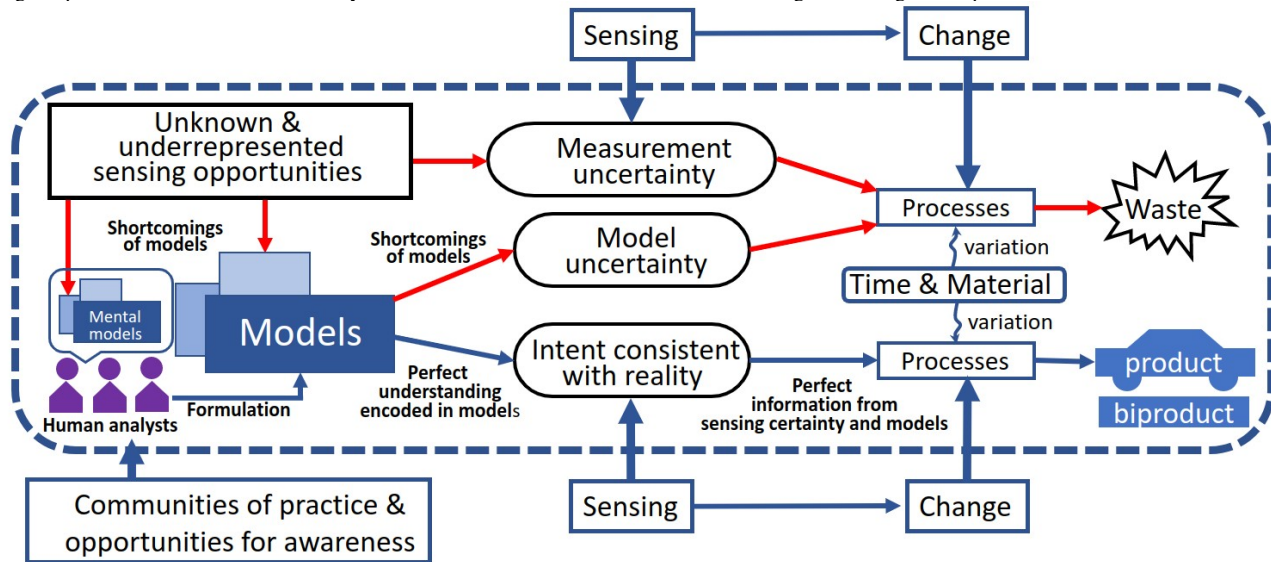
*Joint cognitive work* (JCW) is cognitive work (e.g., planning, perceiving, deliberating) in which human and AI agents interact on cognitive tasks. Precedents to the notion of joint cognitive work discussed in this article include (i) Checkland’s notion of a *learning system* described in the Soft System Methodology [7], (ii) works about establishing common ground [8], [9] and (iii) cognitive systems engineering [10]. More effective than viewing manufacturing’s future as a disparate collection of efforts to automate, one can conceive of a future where some cognitive tasks are performed as JCW. The frontier of the effort to automate might then be pursued as JCW to methodically refine models and implement new sensing. That is the perspective on human-centered AI (HCAI) [11] and management science presented in this article. Many cognitive tasks could be performed as JCW with models. This is becoming increasingly obvious with the emergence of large language models [12] (LLMs) to perform programming tasks. The article discusses some foundational concepts using as an example the joint formulation of production scheduling solutions.

The cognitive work of manufacturing has evolved over many years. Manufacturing and similar complex system engineering could benefit from a new paradigm about how these systems are to be adapted; this is discussed in the following section. Following that discussion, the article considers (enumerated by section number) (ii) the roles of situation awareness [2], scientific explanation [13], [14], models, and metamodels in complex systems, (iii) a framework for manufacturing HCAI called *theory flexecution*, (iv) an example use of the framework to formulate solutions for manufacturing production scheduling, (v) the challenge of developing a measurement science for explanation suitable to the new paradigm, and (vi) assessing where JCW is most feasible.

## II. WHAT HAS CHANGED? WHAT HASN’T?

The way we formulate models for manufacturing has changed greatly; the fact that we do has not. In 1832, Charles Babbage, known best for his work on the Analytical Engine, wrote an insightful book on manufacturing in which he said “*When each processes, by which any article is produced, is the sole occupation of one individual, his whole attention being*

Fig. 1. Models can be viewed as causes in causal chains. Two such chains are depicted, one leading to product, the other to waste. The goal of analysts is to discover assignable causes [15] and encode them in models to control production, thus minimizing the flow to waste. In practice, the system under study may concern a single link as depicted, or a network of such links. The latter might represent a supply chain or manufacturing process chain. The dotted-line rectangle represents the isolation of the analysts' awareness from the environment, its knowledge, and exogenous phenomena.



devoted to a very limited and simple operation, improvements in the form of his tools, or in the mode of using them, are much more likely to occur to his mind, than if it were distracted by a great variety of circumstances.” As this suggests, it was recognized early that the division of labor, making work more routine, made “working smarter, not harder” increasingly possible. What was missing at the time were the abilities to communicate, promulgate, and systematize advancements in practice. About one hundred years after Babbage, Walter Shewhart’s book *Economic Control of Quality of Manufactured Product* [15] answered this need by formalizing an actionable, mathematical, language for production quality. Increasingly, from the time of Shewhart’s work forward, the means to competitiveness has been the formalization and refinement of the models driving manufacturing. Figure 1 depicts a viewpoint on the use of models that is as relevant today as it was in Shewhart’s time.

Driven by ubiquitous sensing, interconnection, and computing power, manufacturing is currently undergoing a Fourth Industrial Revolution, referred to as Industry 4.0 [16]. The capabilities that sparked the revolution have allowed many more concerns to be taken into account in decision making. Further, sophisticated software engineering techniques have eased the cognitive burden of modeling and allowed the composition of large-scale simulations from elementary component models. However, what has not yet emerged is model- and software-orientation in quality programs and management science. The Six Sigma<sup>1</sup> quality program [17], a popular management initiative in manufacturing, is a case in point [18], [19]. The cornerstone of Six Sigma is problem solving by a

process described as *define, measure, analyze, improve, and control* (DMAIC). DMAIC does not take advantage of the fact that the models driving manufacturing are increasingly formal, rather than mental, and of increasingly greater scale. Software-orientation in modeling, the fact that models are now complex composite structures, adds evolutionary and cognitive dimensions to problem solving lacking in the quality programs.

The evolutionary dimension is lacking in the quality programs in the sense that those programs are insensitive to (1) the need for flexibility in problem-solving method and goal-setting, [20] (2) the system’s technology readiness level, [21], (3) improvements in decision support technology such as simulation, and (4) changes in corporate culture and capability, for example, those owing to mergers and acquisitions. From the cognitive perspective, the quality programs do not take advantage of opportunities afforded by (a) widely prevalent, web-assisted communities of practice, and (b) the model as a public “knowledge artifact”, facilitating communities of practice and constructionist learning.

Further, Six Sigma provides virtually no support for efficient system diagnosis [17]. This may be owing to Six Sigma treating systems as black boxes understood only through exploratory data analysis where causal inference [22] might be more effective [23]. In contrast, simulations and digital twins provide domain-specific knowledge facilitating diagnosis. Work in immersive analytics suggests that the tasks of foraging data and generating hypotheses can be shared among human and machine agents [24].

*Dynamic capability*, an established concept in management science, provides a foundation for studying methodical improvement more comprehensively [25]–[27]. Winter [25] describes dynamic capability as the ability to extend, modify, or create the ordinary capabilities of routine activities (activities that allow the firm to “make a living” in the short term).

<sup>1</sup>Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Zollo & Winter [28] note three learning mechanisms used in a firm's effort to create dynamic capability. (1) *Experience accumulation* occurs through participation in routine activities. (2) *Knowledge articulation* is macrocognitive sense-making in which participants express their opinions, engage in constructive confrontations, and challenge each other's viewpoints. (3) *Knowledge encoding* is the effort to formalize the best of what has been articulated, including its representation in standards, manuals, and models. Further, they argue that articulation and encoding are increasingly effective relative to tacit experience accumulation (a) among tasks that are less frequently executed, (b) when task experiences are more diverse, and (c) where uncertainty exists about the causality operating in the task.

Romme et al. [29] describe the effects of deliberate learning and dynamism of the business environment on the ability to change operating routines. Using a system dynamics model, they note how knowledge articulation and knowledge encoding affect the ability to change operating routines. They argue that repeated knowledge articulation unveils causal ambiguity and raises the level of awareness of the effectiveness of the firm's processes. However, both knowledge acquisition and knowledge encoding require time that could be used effectively to interact with operating routines (experience accumulation). Further, though tools created through knowledge encoding may have utility, they can, especially in dynamic environments, result in detrimental inertia.

### III. METAMODELS AS CONVERSATION STARTERS

Engineers refine models as a means to improve outcomes. There is a limit, however, to how much a model can be refined given only the parameters from which it was originally formulated. *Theory* is defined as a hypothesis about the relationship between a collection of models and reality [13], [30], [31]. Dissatisfaction with the operation of one's production facility, for example, stems from one's theory about it (what could be called the *theory gap*); it is very much a matter for human agency.

Today there are hundreds of "languages" including formal models and protocols for communication and quality control used in most technical domains. Among the more technical models are *metamodels*, models of modeling languages themselves [32]. A *structural metamodel* is a model of a modeling language providing sufficient detail to serve as a storage form for the subject models. Structural metamodels provide schematic information about the represented models. The most prevalent examples of structural metamodels are the abstract syntax trees (ASTs) of programming languages. These are not unlike the sentence diagrams used in teaching the grammar of natural languages. However, designed for use in compilers, ASTs may not organize the model content to a form useful to understanding the problem the user is trying to solve. In contrast, metamodels purpose-built for domain-specific languages (DSLs) can highlight the modeling language's relationship to the user's domain problem.

Structural metamodels provide an opportunity for joint human/machine work in formulating and refining models. Joint work, whether it includes machine agents or not, requires that

the parties establish *common ground* [33], such as agreement on the meaning of terms. Illustrative of establishing common ground is the skill professors might exercise during office hours. In office hours, students visit to discuss disparate problems; the professor needs to quickly grasp the essence of the student's question. Let's suppose that the professor teaches a class in production scheduling using a combinatorial optimization solver like MiniZinc [34]. In establishing common ground on a student's MiniZinc problem, the professor would do well to start with the *index sets* the student defined in her model. (See Lines 1 and 2 of Figure 2.)

In program comprehension, features of a program useful to understanding its plan and organization, such as index sets in MiniZinc, are called *beacons* [35]. Index sets can serve as beacons because they typically enumerate things relevant to the real-world problem domain. In production scheduling, the index sets often enumerate production resources, input materials, and work to be performed. Beacons provide a strategy for machine agents to flesh out a metamodel with information about the human's effort to formulate a model, enabling a "conversation" about that effort.

Metamodels, with the help of program comprehension, provide the opportunity for joint human/machine work on models. Figure 3 depicts an architecture for joint model refinement. The bottom region of the figure represents the requirement for expert knowledge of analytical tools used in decision processes. The top region represents knowledge of situation awareness (SA). The technical constraints of model formulation include one's ability to sense the environment and to apply the language of the model's encoding. The middle region of the figure depicts Klein's Flexecution [36] adapted for use with formal models. Flexecution describes the cognitive process by which goals can change based on discoveries made while executing a plan [36]. Klein suggests that in complex settings flexecution is the norm and plans conceived ahead of time and still usable when needed are the exception. *Theory flexecution* is flexecution where the goal is to improve a theory, the relationship between models and reality.

### IV. COGNITIVE DESIGN PATTERNS FOR JOINT FORMULATION

Some of the ideas of joint cognitive work with models outlined above were explored in example usage that formulated production scheduling solutions (solver programs) for apparel manufacturing. The solution used a MiniZinc solver embedded in Python Jupyter notebooks [37]. In a field study, two analysts skilled in production scheduling and the firm's enterprise resource planning (ERP) system worked with the author to transition from a spreadsheet-based process to one that uses the notebook. Typical of apparel manufacturing, sewing is the bottleneck process. Schedules determine the number of sewers of various skill levels that should populate each of the production lines each week to best meet customer orders.

To establish common ground in joint work, it is useful to start with things familiar to most participants. In this case,

spreadsheets of sales orders contain concepts all of which are familiar to the analysts. The machine agent attached to a Jupyter notebook studied spreadsheets loaded into the notebook and made predictions about what is intended by each column or row of data. In this example, Bayesian abductive inference [3] was applied to hypothesize how data and variables are being used and, ultimately, about the general plan and level of completion of the notebook. As the machine agent calculates the most probable explanation (MPE) of how a plan goal is being achieved, it can verify its interpretation with the user. For example, the machine agent might not be able to find direct reference to the notion of job, but it might find three columns in a spreadsheet that look like, respectively, a date, a quantity, and a product type. Bayesian abductive inference might infer that the most probable notion of jobs is provided by 3-tuples of those elements.

The range of what need be anticipated in joint work with models can be significantly reduced in some applications; production scheduling is one such application. First, production scheduling involves a predictable set of activities; every production scheduling problem involves describing (1) work to be performed, (2) resources and their fitness to perform various types of work, and (3) preferences regarding how the resources should be applied to the work. Second, most scheduling problems can be described using constraints and goals. Thus, a declarative domain-specific language such as MiniZinc can be used to describe the scheduling problem. Declarative languages describe what is required, not how what is required is achieved. Though the field study machine agent examined both Python language and MiniZinc text, the declarative nature of MiniZinc made plan recognition from that more tractable than from the Python code.

To foster common ground about constraint-based optimization, analysts can be taught a sentence-based cognitive design pattern [38]. Specifically, combinations of columns in spreadsheets can be viewed as providing sentences describing the scheduling context. Similarly, certain sentence patterns represent the scheduling actions that are part of the resulting schedule. For example, in the apparel example there are three scheduling action sentence types:

- 1) *Production on product ?p begins on week ?wstart.*
- 2) *Production on product ?p stops on week ?wstop.*
- 3) *While product ?p is being sewn, ?n sewers are on its production line.*

The set of all possible scheduling action sentence sets (a

powerset) can be conceived as sets of sentences completed with these variables filled by data. The problem of formulation can then be viewed as one where, first, sentences about input data are related to sentences about the scheduling actions, and then the sentence sets are winnowed by removing contradictions. For example, an entire sentence set is invalid if it contains sentences implying that a job finishes before it starts. Identified contradictions spur the specification of MiniZinc constraints. The notebook, used with the powerset abstraction, can be viewed as an expeditious approach to some of the same characteristics sought in low-code environments [39], [40].

Beneficial socio-technical effects can be credited to the use of the notebook in the field study; the notebook served as a tool for constructionist learning. Further, having learned to be effective with computational notebooks, the analysts went on to develop a complementary notebook for forecasting the size of the workforce, a capability particularly useful during the pandemic. The notebook also proved to be useful in conversations with other stakeholders, such as plant managers. The notebook, as a public object, serves to integrate system elements.

## V. THE ROLES OF EXPLANATION

The previous section describes a simple example of JCW. In that, explanations provided by program comprehension using the MiniZinc metamodel focus on goals and are mechanistic in describing means. “Mechanisms are entities and activities organized such that they are productive of regular changes from start or set-up to finish or termination conditions” [41]. Mechanistic explanations explain *why* by explaining *how* [42]. However, in as far as the individual steps of *how* (the causes) are uncertain, mechanistic explanation is unconvincing. Not all the explanation that would be useful in joint work is mechanistic.

JCW needs additional modes of explaining. Manufacturability problems, for example, often involve a search for causes (not just correlation) through experimentation. The method of explanation most relevant to such investigations is interventionism through experimentation and counterfactual reasoning [22], [43], [44]. The essential idea behind interventionism, and what distinguishes it from quality programs such as Six Sigma, is that under certain conditions, intervention in the process under study reveals a cause. Roh et al. [45] and Witherell et al. [46] define metamodels for experimental exploration of an advanced manufacturing process. These

Fig. 2. A MiniZinc formulation of a simple resource allocation problem (not the apparel scheduling problem), assigning  $n$  tasks to  $n$  workers minimizing the cost of getting all the tasks done. Lines 1 and 2 are index sets. Line 3 defines the cost of each worker to do each task. Line 4 describes the form of solutions. The simple declarative syntax, the choice of meaningful variable names, and the use of natural language annotations increase the likelihood that a machine agent could recognize what the analysts are seeking to achieve and how they are going about it.

---

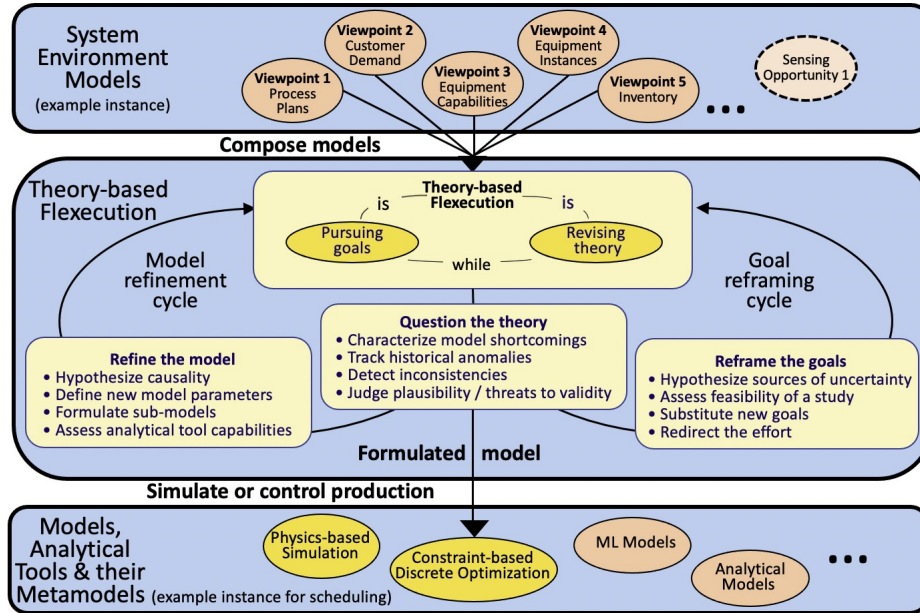
```

1 set of int: Workers = 1..n;           % An index set, the number of workers...
2 set of int: Tasks   = 1..n;           % An index set, the number of tasks, same number as workers.
3 array[Workers, Tasks] of int: cost;   % Worker i does Task j at cost[i,j].
4 array[Workers] of var Tasks: doesTask; % 'var' thus this is a decision variable found by the solver.
5
6 constraint alldifferent(doesTask);     % is a common constraint in such solvers; no two workers do the same task.
7
8 solve minimize sum (w in Workers) (cost[w,doesTask[w]]); % Minimize the total cost.

```

---

Fig. 3. Theory flexecution involves refinement of models in consideration of the system environment, theory, and tool capabilities. This center portion of the diagram is adapted from Klein’s Flexecution [36] to emphasize the roles of models, metamodels, and theory.



metamodels federate submodels of the individual physical processes (heating, melting, thermal conductivity) of the subject additive manufacturing process. Fitting to this purpose, these are not structural metamodels but knowledge graphs built from the Web Ontology Language (OWL). Using knowledge from the manufacturer’s own experimentation with the process, as well as that of the literature, such metamodels help with the tasks “Question the theory” and “Refine the model” as depicted in Figure 3. These are metamodels for applying an interventionist mindset in theory flexecution.

A third viewpoint on explanation, unificationism, is of less obvious value to engineering than it is to science. *Unification* highlights connections and common patterns in a body of knowledge (or “explanatory store”); it seeks the best tradeoff between the number of patterns of derivation used and the number of conclusions that can be generated [47]. There is obvious value in science to finding common patterns and confirming evermore general hypotheses; that could be taken as the purpose of science. The unificationist view is also valuable to JCW owing to its inherent need to weed out irrelevant elements of an argument. (Kitcher, who first described unificationism, contrasts the differences between explanations of (1) how salt dissolves in water, and (2) how salt “hexed by a magician” does it [47].) Unification is a key activity in sense-making and immersive analytics [48] [24].

Explanation has a role in organizing activity towards dynamic capability. The three modes of explaining discussed each contribute differently to flexecution. Mechanistic, interventionist, and unificationist explanations reveal or argue for, respectively, a mechanism, a cause, and a uniting principle.

There is also the matter of classifying and organizing the individual propositions of an explanation. This can help identify the nature of the theory gap, the difference between the model and the observed reality. One method for classi-

fying scientific explanations, demonstrated by Overton [49], involved classifying sentences of a collection of journal papers according to how each sentence bridges five notions: theory (meaning here the “laws” of a domain of investigation), model, kind, entity, and observation. Each sentence is associated with a class defined by its participation in two of the five notions (thus the 25 classes theory-theory, theory-model, model-kind, etc.). Recent work in natural language processing such as participant, intervention, comparator, and outcome (PICO) extraction automates a similar kind of classification [50].

Ultimately, a measurement science for explanation in joint work will be needed. How do explanations facilitate joint work? Theory flexecution is a methodical way to discover and bridge the gap between models and observed reality. Discovery, however, is not necessarily what motivates explanation in areas outside of joint work with models. For example, efforts towards explainable AI (XAI) [33] typically focus on trustworthiness and accountability to the end user of a model [51], [52]. Compared to JCW with models, XAI is more a situation in which experts are developing models for a disjoint population of users. Despite these differences, there are certainly areas where the goals of XAI and explanation for JCW intersect. For example, some measurement techniques for XAI employ a teacher/student paradigm; this may be useful for assessing explanation in JCW also [53].

Finally, there are broad implications for systems engineering were it to accommodate explanation as a distinct kind of communication. Communication of *information* serves the ordinary goal of eliciting behavior from other agents; communication of information allows components to act as a system [54]. In JCW with models, explanations are communicated to further *understanding* [55]. Presumably, understanding changes behavior in some enduring yet hard-to-predict ways. A possibly useful distinction between communicating

information and communicating to further understanding is that the latter might be viewed as concerning type causality [56] whereas the former only concerns actual causality.

## VI. WHERE IS JOINT COGNITIVE WORK MOST FEASIBLE?

We have discussed a broad range of topics including meta-models, problem solving, dynamic capability, compositionality, declarative formulation, and the constructionist notion of public object. We have argued that explanation plays key roles in enabling joint work with models. To summarize this discussion, the following five characteristics are cited as contributing to the likelihood that a given investigation is fit for joint human/machine theory flexecution:

**unitary public object:** A *public object* reifies thought into some technical object (such as a Jupyter notebook) that can be shared among participants. A *unitary* public object is one that provides a comprehensive viewpoint by which the necessary cognitive work can be performed.

**community-forming:** This assesses the possibility of a community of practice (intramural or otherwise) emerging from use of a public object. On-line classes for MiniZinc is an example of such a community of practice.

**multiple explanatory modes:** Where more than one mode of explaining appears necessary, this assesses how effectively the modes and public objects might integrate.

**compositional metamodel:** Compositionality allows units of functionality (such as simulations) to interoperate despite differences in their internal logic. Where a unitary public object is not feasible, compositionality offers an alternative solution. Compositionality is increasingly valuable in settings where it is uncertain how an investigation might evolve.

**declarative metamodel:** Declarative metamodels like MiniZinc's abstract away messy detail about how a goal is to be achieved. Thereby, declarative metamodels present an opportunity for a machine agent to provide this expertise. Declarative metamodels also facilitate machine agent plan recognition. Generally speaking, the more declarative the metamodel, the better the opportunity for theory flexecution as JCW.

## VII. CONCLUSION

Increasingly, models mediate the relationship between humans and the resources of complex systems. As the models become more complex, we risk losing touch with the rationale for their complexity, and therefore the ability to adapt them. At the highest level, we identify an opportunity to re-examine the roles of models through the lens of management science. Theory flexecution provides a road map for the methodical use of models, metamodels, and analytical tools. In theory flexecution, analysts negotiate goals mindful of model shortcomings, measurement uncertainty, and the capabilities of the analytical tools used. Little by little, engineers are introduced to a sophisticated tool and a link from their requirements to a solution is established.

Consensus is firm that manufacturing's future involves increasing levels of automation. To some, increased automation may suggest diminished roles for humans. It is with some

irony then that we recognize that human roles in the evolution of manufacturing systems seem assured; the challenge lies in finding roles for machine agents in system evolution. The possible future described is a human-centered one, where human agency continues to direct the processes of discovery. Shneiderman [57] notes that "Human curiosity and desire to understand the world means that humans are devoted to causal explanations, even when there is a complex set of distant and proximate causes for events." Explanation's role in theory flexecution could keep human agency at the center of manufacturing's future. Theory flexecution is directed at some of the same goals as manufacturing's quality programs. The quality programs have been refined over many years, but they have yet to take advantage of advances in simulation, modeling languages, and software engineering. Theory flexecution is a means to address these shortcomings and provide the evolutionary and cognitive dimensions of dynamic capability. The potential benefits of theory flexecution include continuous improvement, knowledge retention, training in analytical skills, and resilience.

As described in the paper, a key role of the machine agent is to guide formulation. This invites comparison to emerging generative AI methods of producing models in formal languages. Compared to state-of-the-art generative AI for direct formulation of models, this work emphasizes a long-running relationship with engineers where rationale for the current formulation is provided. It is conceivable that more powerful generative AI may someday subsume the low-level functions of keeping models consistent with requirements. [More here about implementing a sensing opportunity.]

Our near-term plans include exploring the capabilities of large language models to analyze analysts' descriptions of scheduling problems. Technology trends such as low-code tools and executable notebooks are providing some of the building blocks for near-term progress. Long-term goals include developing a measurement science for the use of explanation in joint work.

## REFERENCES

- [1] Christian Weckenborg, Patrick Schumacher, Christian Thies, and Thomas S. Spengler. Flexibility in manufacturing system design: A review of recent approaches from Operations Research. *European Journal of Operational Research*, page S0377221723006781, September 2023.
- [2] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.
- [3] Gita Sukthankar, Robert P. Goldman, Christopher Geib, V. Pynadath, David, and Hung H. Bui. *Plan, Activity and Intent Recognition*. Morgan Kaufmann, Amsterdam, 2014.
- [4] Gary Stager. Towards a pedagogy of online constructionist learning. *8th IFIP World Conference on Computers in Education, WCCE 2005*, July:1–9, 2005.
- [5] Seymour Papert and Idit Harel. Situating Constructionism. In *Constructionism*. Ablex Publishing Corporation, 1991.
- [6] Carolyn E Psenka, Kyoung-Yun Kim, Gul Kremer, and Karl R Haapala. Translating Constructionist Learning to Engineering Design Education. *Journal of Integrated Design and Process Science*, 21(2):3–20, 2017.
- [7] Peter Checkland. Soft Systems Methodology: A Thirty Year Retrospective. *Systems Research and Behavioral Science Syst. Res.*, 17:11–58, 2000.
- [8] Robert Wray, James Kirk, and Jeremiah Folsom-Kovarik. Improving Common Ground in Human-Machine Teaming: Dimensions, Gaps, and Priorities. In *13th International Conference on Applied Human Factors and Ergonomics (AHFE 2022)*, 2022.

- [9] Gary Klein, David D. Woods, Jeffrey M. Bradshaw, Robert R. Hoffman, and Paul J. Feltovich. Ten challenges for making automation a "team player" in joint human-agent activity. *IEEE Intelligent Systems*, 19(6):91–95, 2004.
- [10] D. Woods, David and Erik Hollnagel. Chapter 6: Functional Synthesis, Laws, Design. In *Joint Cognitive Systems: Patterns in Cognitive Systems Engineering*, pages 55–59. Taylor & Francis, 2006.
- [11] Mark O. Riedl. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33–36, January 2019.
- [12] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, June 2023.
- [13] Ronald N. Giere. *Explaining Science: A Cognitive Approach*. University of Chicago Press, Chicago, 1988.
- [14] Kareem Khalifa. *Understanding, Explanation, and Scientific Knowledge*. Cambridge University Press, Cambridge, 2017.
- [15] A. Shewhart, Walter. *Economic Control of Quality of Manufactured Products*. John Wiley & Sons, 1931.
- [16] Klaus Schwab. *The Fourth Industrial Revolution*. World Economic Forum, 2016.
- [17] Jeroen De Mast and Joran Lokkerbol. An analysis of the Six Sigma DMAIC method from the perspective of problem solving. *International Journal of Production Economics*, 139(2):604–614, 2012.
- [18] Mikel Harry and Richard Schroeder. *Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations*. Doubleday Business, 1999.
- [19] Ronald J.M.M. Does and Albert Trip. The Impact of Statistics in Industry and the Role of Statisticians. *Austrian Journal of Statistics*, 30(1):7–20, 2001.
- [20] Kevin Linderman, Roger G Schroeder, Srilata Zaheer, and Adrian S Choo. OPERATIONS Six Sigma : A goal-theoretic perspective. *Journal of Operations Management*, 21:193–203, 2003.
- [21] European Association of Research and Technnology Organizations. The TRL Scale as a Research & Innovation Policy Tool, EARTO Recommendations. Technical Report April, Brussels, 2014.
- [22] Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference in Statistics: A Primer*. John Wiley & Sons, Ltd, 2016.
- [23] Wallace J. Hopp and Mark L. Spearman. *Factory Physics, Third Edition*. Waveland Press, Inc., Long Grove, IL, 2008.
- [24] Richard Skarbez, Nicholas F. Polys, J. Todd Ogle, Chris North, and Doug A. Bowman. Immersive Analytics: Theory and Research Agenda. *Frontiers in Robotics and AI*, 6(September), 2019.
- [25] Sidney G. Winter. Understanding dynamic capabilities. *Strategic Management Journal*, 24(10 SPEC ISS.):991–995, 2003.
- [26] Marc Sniukas. *Business Model Innovation as a Dynamic Capability: Micro-Foundations and Case Studies*. Springer, Berlin, 2020.
- [27] Constance E Helfat, Sydney Finkelstein, Will Mitchell, Margaret A. Peteraf, Harbir Singh, David J. Teece, and Sidney G. Winter. *Dynamic Capabilities: Understanding Strategic Change in Organizations*. Blackwell Publishing, Malden, MA, 2007.
- [28] Maurizio Zollo and Sidney G. Winter. Deliberate Learning and the Evolution of Dynamic Capabilities. *Organization Science*, 13(3):339–351, 2002.
- [29] Georges Romme, Maurizio Zollo, and Peter Berends. Dynamic capabilities, deliberate learning and environmental dynamism: A simulation model. *Industrial and Corporate Change*, 19(4):1271–1299, 2010.
- [30] Bas C van Fraassen. *The Scientific Image (Clarendon Library of Logic & Philosophy)*. 2009.
- [31] Frank Hindriks. False Models as Explanatory Engines. *Philosophy of the Social Sciences*, 38(3):334–360, 2008.
- [32] Ed Seidewitz. What Models Mean. *IEEE Software*, 20(5):26–32, 2003.
- [33] Mark Stefik. Explainable AI: An Overview of PARC's COGLE Project with DARPA. *PARC*, 2021.
- [34] Peter J. Stuckey, Kim Marriott, and Guido Tack. MiniZinc Handbook Release 2.2.1. Technical report, Creative Commons, 2018.
- [35] Sean Stapleton, Yashmeet Gambhir, Alexander LeClair, Zachary Eberhart, Westley Weimer, Kevin Leach, and Yu Huang. A human study of comprehension and code summarization. *IEEE International Conference on Program Comprehension*, pages 2–13, 2020.
- [36] Gary Klein. Flexecution as a paradigm for replanning, Part 1. *IEEE Intelligent Systems*, 22(5):79–83, 2007.
- [37] Project Jupyter. Jupyter Project. <http://jupyter.org/about.html>, 2017.
- [38] Christopher R. Hale and Vincent Schmidt. Cognitive design patterns. *Proceedings of the 2008 International Conference on Software Engineering Research and Practice, SERP 2008*, pages 611–616, 2008.
- [39] Arun S. Maiya. Ktrain: A Low-Code Library for Augmented Machine Learning. *arXiv preprint:2004.10703*, 10703, 2020.
- [40] Gwendal Daniel, Jordi Cabot, Laurent Deruelle, and Mustapha Derras. Xatkit: A Multimodal Low-Code Chatbot Development Framework. *IEEE Access*, 8:15332–15346, 2020.
- [41] Peter Machamer, Lindley Darden, and Carl F Craver. Thinking about Mechanisms. *Philosophy of Science*, 67(1):1–25, 2000.
- [42] William Bechtel. Enriching the Strategies for Creating Mechanistic Explanations in Biology. In *Visions of Cell Biology*. University of Chicago Press, 2017.
- [43] James Woodward. *Making Things Happen*. Oxford University Press, Oxford, 2003.
- [44] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2001.
- [45] Byeong-min Roh, Soundar R T Kumara, W. Simpson Timothy, Pan Michaleris, Paul Witherell, and Ibrahim Assouroko. Ontology-based Laser and Thermal Metamodels for Metal-based Additive Manufacturing. pages 1–8, 2016.
- [46] Paul Witherell, Shaw Feng, Timothy W. Simpson, David B. Saint John, Pan Michaleris, Zi-Kui Liu, Long-Qing Chen, and Rich Martukanitz. Toward Metamodels for Composable and Reusable Additive Manufacturing Process Models. *Journal of Manufacturing Science and Engineering*, 136(6):061025, 2014.
- [47] Philip Kitcher. Explanatory Unification and the Causal Structure of the World. In *Scientific Explanation*, pages 410–505. University of Minnesota Press, Minneapolis, 1989.
- [48] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis*, 2005(January):2–4, 2005.
- [49] James Alexander Overton. *Explanation in Science*. PhD thesis, University of Western Ontario, 2012.
- [50] Shifeng Liu, Yifang Sun, Bing Li, Wei Wang, Florence T. Bourgeois, and Adam G. Dunn. Sent2Span: Span Detection for PICO Extraction in the Biomedical Text without Span Annotations. pages 1705–1715, 2021.
- [51] Filip Karlo Dosiilovic, Mario Brcic, and Nikica Hlupic. Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, pages 210–215, 2018.
- [52] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for Explainable AI: Challenges and Prospects. (December), 2018.
- [53] Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating Explanations: How much do explanations from the teacher aid students? *arXiv:2012.00893v2*, 2021.
- [54] Gerald R. Miller. On Defining Communication: Another Stab. *Journal of Communication*, 16(2):88–98, 1966.
- [55] Peter Achinstein. *The Nature of Explanation*. Oxford University Press, Oxford, 1983.
- [56] Y. Halpern, Joseph. *Actual Causality*. The MIT Press, Cambridge, MA, 2016.
- [57] Ben Shneiderman. Human-Centered AI. *Issues in Science and Technology*, (Winter):56–61, 2021.