Visual Computing Challenges

Visual Analytics for Early-Phase Complex Engineered System Design Support

Rahul C. Basole, Ahsan Qamar, Hyunwoo Park, Christiaan J.J. Paredis, and Leon F. McGinnis • Georgia Institute of Technology

Complex engineered systems (CESs) can be found in a range of contexts. In today’s industrial settings, two types of CESs are particularly pervasive. Cyber-physical systems (CPS) are smart networked systems with embedded sensors, processors, and actuators that are designed to sense and interact with the physical world including human users. Cyber-physical production systems (CPPS) integrate CPS technologies with smart logistics systems and production facilities, thereby enabling highly flexible and configurable manufacturing.

The design and production of such CESs requires consideration of many stakeholder concerns ranging from manufacturing and the supply chain to finance and corporate management. These concerns can be diverse and frequently conflict with each other. Seamless access and communication of relevant information and interaction between stakeholders are thus critical to facilitate effective decision making and achieve timely resolution across all phases of CES design and production. What complicates matters is that the design, manufacturing, and supply chain enterprise functions can be geographically dispersed and involve people with many different backgrounds and mental models that can influence their interpretation of the presented information.

Model-based systems engineering (MBSE) has emerged as one promising approach for dealing with the collaborative, distributed design and production of CESs, and it advocates the formalized application of modeling to support systems engineering activities throughout the product life cycle. The focus is not only on the aspect of the system that a model captures, but also on whether the available knowledge supports good communication and, as a result, good decision making. MBSE requires making appropriate choices about the level of abstraction, modeling formalisms, interactive model representations, and model verification and validation to both attend to the essential complexity of the system and minimize the accidental complexity that will otherwise result from inappropriate selections.

With the growing complexity of MBSE models and overlapping stakeholder concerns, however, ensuring a comprehensive understanding and analysis of the underlying information is becoming increasingly challenging. Moreover, technological advances have led to an increasing scale and availability of digital data in CES design. Appropriate tools are required that consider various sources of information about the CES and can identify, extract, and curate the information to suit the needs of various decision makers. These new macroscopic tools and analysis

Visual analytics tools can provide interactive discovery, exploration, and understanding of real-world complex engineered systems (CES). The proposed tool, which focuses on the early design phase, can help users perform routine CES design analysis tasks and offer stakeholder-specific visual representations of complex design models.
Visual Computing Challenges

Related Work in Visual Analytics Tools

A key motivation of our tool is to provide novel and possibly complementary visual analytic capabilities to existing model-based systems engineering tools. Specifically, our tool aims to provide linked visual representations with dynamic interaction techniques of complex engineered systems (CES) data. Following our comprehensive requirements analysis, we found it important that such a system allows the user to not only interactively identify the type of information that they are interested in, but also modify how the information is presented to them.

There are some CES design tools that enable creating links between information from heterogeneous sources and provide an overview of how design elements are interconnected. IBM’s Rational Engineering Lifecycle Manager (www-03.ibm.com/software/products/en/rationallife) and Mentor Graphic’s Context SDM (www.mentor.com/products/sm/context-sdm) are two examples that use Open Services for Lifecycle Collaboration as a basis to create links between information from heterogeneous modeling sources. However, most existing visualization tools in manufacturing focus on providing graphical 3D representations of the actual physical system. In the aerospace industry, FlyThru1 and Massive Modeling Visualization2 are two examples of 3D aircraft modeling tools enabling the incorporation of multiple levels of design detail.

In contrast, the focus of our work is on abstract models that are developed in the early stage of design rather than on 3D data. Most of these models are captured in a block-diagram type formalism, where the system of interest is abstracted as a network of elements connected through ports, for example, in UML/SysML, Simulink, Modelica, Visio, and so on. Our focus is thus to use the data from these type of sources to create a modeling view for the decision maker based on their concerns and the preferred mode of presentation. To the best of our knowledge, no such visual analytic tool is currently available to system engineers. Currently the only way to present and explore their model visually is through diagrams, which are static and prone to visual cluttering. Moreover, as diagrams are created a priori, the only way to incorporate information from multiple sources into a diagram is through model transformations, which also leads to highly constrained visualization and interaction capabilities.

References

methods are consequently becoming a requirement for the manufacturing enterprise.

Visual analytics, the fusion of information visualization with analytics capabilities, promises to move beyond the common ways of consuming information toward interactive discovery, exploration, and understanding of CESs.3 The goal of CES visualizations is to leverage the human visual system’s highly tuned ability to see patterns, spot trends, and identify outliers in data.4 Well-designed visualizations can improve comprehension, memory, and decision making as well as engage diverse stakeholders in exploration and analysis.5 In addition, visualizations can be customized through the careful analysis of data, user demands, and context of the decision problem, thus enabling improved communication and accessibility of information.

Visual analytics tools can advance an enterprise’s existing system engineering capabilities. Visualizations currently available in existing MBSE design and analysis tools do not serve the purpose of multiple decision makers because they are built for a particular stakeholder with a certain domain knowledge, and they are incapable of visualizing data from other sources that do not conform to the abstract syntax of the modeling language that the tool supports. Diagrams are the most prevalent technique used in current MBSE tools. However, these diagrams are static, do not scale well with depth of information, and require substantial effort to evolve over time. Moreover, these representations do not take into account the influence of data interdependencies (that is, interviewpoint relations). The design and development of an effective visual analytic tool is therefore not without challenges. (See the “Related Work in Visual Analytics Tools” for earlier tools and approaches.)

This article reports on our ongoing experiences in developing visual analytics tools for real-world CESs. Our work focuses on the early design phase during which a large design space is explored, poor alternatives are pruned, and valuable alternatives are considered further.

CES Design Challenges
A common way to understand the complexity of a CES is to examine it from different viewpoints, thereby partitioning the overall problem into subproblems. A viewpoint is defined by a variety of factors, including the concerns of interest, level of abstraction, modeling formalism, and context.6 Although each viewpoint aids in dealing only with a limited set of concerns, the partitioning does not guarantee independence between viewpoints, and overlaps and conflicts (between stakeholder concerns) could exist. Decision makers must therefore take into account overlaps to establish the right context for their decision problem. A decision maker may need to switch between viewpoints and evaluate how the viewpoint-level decisions influence the system-level decisions.
Visualizations catering to a particular viewpoint could thus only provide partial insights and not completely serve the decision maker’s purpose because they potentially disregard the influence of other viewpoints. Dealing with multiple system modeling sources, however, is a challenging problem in itself due to distributed development, the heterogeneous nature of tools and modeling formalisms, and the difficulty in achieving integration and interoperability among the design tools and between design and manufacturing tools. For example, the design of a physical artifact influences the manufacturing process used to build it, and the information about current manufacturing interoperability of the underlying data is therefore an essential foundation for an effective visual analytics tool.

Given the variety of stakeholders, there are many different analysis tasks that shape the design and development of a visual analytics tool for CES design. A product designer, for instance, could be interested in exploring the integrated MBSE data repository to visually determine overlaps with other stakeholder concerns. Supply chain managers may be interested in seeing the geographic distribution of suppliers and identifying which suppliers carry a more central role in the supply chain. Other possible analysis examples include corporate decision makers who participate in design reviews with the technical teams or venture capitalists who wish to evaluate the market potential. The visual literacy of these user types can be quite varied and must be carefully addressed when designing effective visual analytics tools.

The predominant way to visualize MBSE data today is through diagrams. The Object Management Group’s Systems Modeling Language (SysML, see www.omg.org/spec/SysML/1.3/), for example, allows the creation of different types of diagrams to capture a system’s structural and behavioral specifications. In this way, these diagrams provide the ability to create different graphical representations of the system specifications. For instance, a block-definition diagram depicts the system hierarchy, whereas a state machine diagrams depicts its behavior. Other modeling languages provide different graphical, primarily diagrammatic representations to attend to a particular stakeholder’s needs—for example, a function modeling language visualizes the function-means tree. However, these diagrams are prone to visual cluttering, are difficult to modify or customize to each user once created, and provide limited interaction capabilities. Figure 1 shows a common diagram of a system model’s structure. Clearly, such a diagram is created for a particular use and context; it is static and does not evolve over time, provides little exploratory flexibility, and cannot be dynamically adjusted for another stakeholder. Therefore, a key motivation of our visual analytics tool is to complement the graphical capabilities of current MBSE tools and provide linked views, interactive capabilities, and support for multiple formalisms.

**Visual Analytics for CES Design Support**

The design and development of visual analytics tools begins with determining which questions to ask (or which tasks to support); identifying the appropriate data; selecting effective visual encodings to map data values to graphical features such as position, size, shape, and color; and providing rich interaction dynamics.6–7

**Requirements**

Following an analysis of common CES early-phase design tasks, we identify five key capabilities that an effective visual analytics tool should exhibit:

---

**IEEE Computer Graphics and Applications**

43
Visual Computing Challenges

- **Provide a macro-to-microscopic perspective.** Stakeholders often want to have an understanding of the entire CES design information space as well as take into account detailed capabilities while making decisions about a product design. The visual analytics tool must therefore provide linked whole-to-part context and interaction at multiple levels of CES design.

- **Accommodate heterogeneous data types.** Relevant CES design data comes from a variety of sources and in many forms, including SysML models, Simulink (see www.mathworks.com/products/simulink/) models, bill of materials, Resource Description Format (RDF) triple stores, CAD drawings, and structured and unstructured textual reports. The visual analytics tool must be designed to accommodate the integration of heterogeneous data sources and provide capabilities to interact with multiformalism data.

- **Facilitate inconsistency discovery and uncertainty management.** One key function of visual analytics tools for CESs is the identification of underlying semantic relationships, uncertainties, and/or inconsistencies in design. The visual analytics tool must therefore facilitate interactive exploration of relationships among design elements, identify misalignments and/or opportunities for integration.

- **Provide temporal views and interactions.** CES design evolves over time and is highly dynamic. The visual analytic tool in CES design must provide event and milestone representations and provide the temporal context of the model state through multiple time snapshots and/or animations.

- **Provide linked stakeholder views.** CES design is a collaborative, multistakeholder activity. Although each stakeholder will have different visualization requirements, some may need to understand how their views fit and interact with other stakeholder requirements. The visual analytics tool must therefore allow for multiple connected views across stakeholders, revealing common touch points and key differences.

**Visualization Techniques**

Visual representations are a fundamental component of human learning and understanding. They enable us to not only communicate information or facts, but enable us to create, assess, and transfer insights, experiences, expectations, and perspectives. By mapping data to visual encodings, visualizations in CES design make the what, why, how, and who explicit. There are a plethora of visual representations available. They range from the simple to the complex. A comprehensive review is beyond the scope of this article, but interested readers are referred to other works for excellent overviews.3-4,8 Given that CES design organization and the relationships of the elements that make up the CES design are of particular interest to early-phase design stakeholders, our discussion of visual representations focuses on three relevant techniques: treemap, force-directed network, and adjacency matrix views.

**Treemap view.** CES design data are hierarchically organized. Stakeholders are often interested in how model elements are grouped and structured. A popular technique to visualize organization is through treemaps (see Figure 2a). A treemap recursively subdivides areas into rectangles. Each rectangle represents a CES model; the size and color of each rectangle can be modified based on relevant attributes. Many treemaps also use padding to emphasize groupings, which facilitates readability, hierarchy detection, and size estimation.
**Force-directed network view.** In addition to the organization of CES design data, stakeholders also want to understand the relationship between the data elements. Questions of interest include the following: Which models are interconnected? Which model is the most central? Which models cluster together? And which model, if removed, would disconnect the model network? Node-link diagrams, where nodes represent models and edges represent relationships between models, are a common and intuitive approach to visualize relationships (see Figure 2b). Node positioning is often determined using force-directed layout algorithms, with the idea that related nodes are pulled (placed) together and unrelated ones are pushed apart.

**Adjacency matrix view.** When networks are large, it is often difficult to discover interesting structure because the sheer number of edges makes it difficult to draw a network without creating a hairball, even with sophisticated layout algorithms. An alternative visualization technique is the adjacency matrix (see Figure 2c). An adjacency matrix is a symmetric table that shows model entities in both rows and columns, and the cell entries can indicate an attribute-based relationship between entities. When adequately reordered, through a technique called block modeling, a matrix can reveal both global and local structures in a CES design network.

**Interactive Dynamics**

Visual representations alone do not suffice for successful analytical discourse of CES design data; it also requires user-friendly, interactive dynamics. Interaction allows stakeholders to explore, make sense, and engage with the underlying dataset, thereby facilitating and accelerating the cognitive reasoning process. Broadly, there are three categories of interaction dynamics that support the fluent and flexible use of visualizations: data and view specification, view manipulation, and analysis process and provenance. Across these categories, we found several interaction techniques of particular relevance for early-phase CES design support: filtering, selecting, navigating, coordinating, and sharing.

Filtering is an essential control in CES visual analytics tools. CES stakeholders frequently want to visualize only a subset of the CES data based on a set of dimensions or criteria, for example, to examine different levels of model parameters or to isolate specific model types. The choice of the appropriate interaction technique is largely determined by the underlying data type. Categorical or ordinal CES data, for instance, can be filtered using radio buttons or checkboxes, scrollable lists, hierarchies, or search boxes. Quantitative or temporal CES data can be filtered using standard or range sliders by specifying endpoints. When dynamically linked to the visualizations, dynamic filters allow rapid and reversible exploration of data subsets.

Selecting a relevant model entity—either for current or subsequent analysis focus—is another important control in CES visual analytics tools. CES stakeholders, for example, may want to select a specific model component, a relationship between models, or a range of entities. Common techniques for selection in visualizations include mouse hover, mouse click, region selection, or brushing. The selection control is closely related to filtering because it can be used to identify or remove objects of interest.

Navigating in visual analytics tools refers to how users traverse the visual information space and commonly follows the information-seeking mantra “overview first, zoom-and-filter, then details on demand.” Given the complexity and scale of CES design data, this approach is particularly applicable. CES stakeholders might begin by taking a broader view of the data—including assessments of key patterns, clusters, and outliers—followed by drill-down and filtering and requesting additional details. This navigation pattern is generally complemented by focus plus context methods, providing a detailed view of the data of interest while providing the overall context. For example, a CES design could focus on SysML model elements while providing the context of how these elements relate to the elements in the Simulink model.

A complete CES design is described by multiple, heterogeneous data sources. A visual analytics tool must therefore be able to provide multiple, coordinated views of these data sources. Coordinated views facilitate comparison and identification of overlaps/inconsistencies in design requirements. For instance, through brushing-and-linking, selecting a model entity in one view would highlight the corresponding entity in all other views. Similarly, multiple different views allows interactive exploration across views and provide complementary CES design perspectives.

CES design is a multistakeholder collaboration activity. A visual analytics tool for CES design must therefore provide annotating, discussing, and sharing capabilities, such as bookmarks, graphical annotations, textual comments, and design export. Moreover, because CES design stakeholders are commonly geographically distributed, visual analytics tools must provide both synchronous and asynchronous collaboration capabilities, including version and access control, activity awareness, and ownership status.
System Design and Implementation

The user interface of our visual analytics tool for CES design support, shown in Figure 3, consists of two main areas: the filter panel and the visualization pane. The filter panel contains options to select model data sources and filter by various node/edge types and their attributes. Changes in any of the filters are dynamically reflected in the visualization. Users can switch between visualization techniques using shortcuts and manipulate the visualization through a range of interaction techniques, including zoom, labels on/off, edges on/off, and hover-over for details on demand. We developed and implemented the interactive user interface using D3.js, an open source JavaScript library that uses digital data to drive the creation and control of dynamic and interactive Web-based visualization (www.d3js.org).

Our Web-based tool draws on multiple heterogeneous system modeling data sources. Models contained in these sources span different domains (for example, requirements management, mechanical CAD, and dynamic analysis) and are described through domain-specific vocabulary or schema. To cope with this, we defined a base vocabulary that serves as a common denominator between multiple domain-specific vocabularies. The advantage of using such an approach is the varying semantic precision while moving from the tool specific vocabulary toward the domain and the base vocabulary. While the tool-specific vocabulary carries greater semantic depth, the base vocabulary is useful for describing the common abstraction between heterogeneous models.

We use a NoSQL graph database (http://jena.apache.org/documentation/serving_data/) as our underlying data repository. Graph databases are particularly suitable for MBSE design data because they do not depend on a rigid schema, as relational databases do, and they allow for time-evolving schemas. Moreover, while relational databases suffer in performance as the number of records in the table increases, graph databases only suffer in performance as the number of connections between nodes in the graph increases. Obviously, for a large-scale CES, such as an aircraft, the datasets could explode in size and scalability could become a challenge.

We represent data from multiple sources in an RDF and use an RDF query language (SPARQL) to query the graph store. To read and write data from multiple system modeling sources, we use a technology called Open Services for Lifecycle Collaboration (OSLC). OSLC takes advantage of the Semantic Web and linked-data principles and expresses each source of data as a set of resources exposed using RDF. RDF facilitates data merging, even if the underlying schemas differ; hence it is well suited for dealing with distributed, heterogeneous, and disparate sources of information commonly found in MBSE contexts.

Illustrative Use Case

We purposely chose an industrial robot as an CES example because it has sufficient complexity and allows us to demonstrate the foundational challenges of designing a CES, including addressing multiple stakeholder concerns and presentation to corporate decision makers during design reviews. (Other commonly known CES examples include an airplane or a spacecraft.) An industrial robot is a CES that, depending on the use context, can be a CPS (for example, one autonomous robot using sensor networks and network control) or be part of a CPPS (for example, a group of autonomous, collaborating, and networked robots). Our 2 DOF (degrees of freedom) robot consists of two mechanical arms actuated through DC motors, with sensors measuring the amount of joint rotation (see Figure 4). The robot acts as an autonomous pick-and-place unit used in an assembly plant. It receives parts from a conveyer belt and delivers the part to a milling machine. All these units are networked with each other and can take user commands or provide data about current operation or configuration. The complete system is thus a good
example of a CPPS. Moreover, the robot is reconfigurable as it can also be used as a drilling or a welding robot.

To ensure that the robot meets the demands of the typical production scenario, the parts processed per minute and the accuracy of the robot’s item pick-and-place functionality are important design concerns. These concerns are addressed through a number of specification and analysis models. While specification models capture the robot’s specification (such as a robot with a pick-and-place gripper), analysis models predict an outcome with a particular specification—for example, position control error or parts per minute when using a (specific) robot with a gripper. Moreover, any reconfiguration requires analyzing the robot’s behavior in coordination with other units, a task requiring reviewing a number of design and analysis models. Based on these characteristics, it is therefore reasonable to argue that the design of our robot is data intensive involving heterogeneous models and tools.

For our use case, we consider two viewpoints: system design and dynamic analysis. We use SysML to support the system design viewpoint, and Simulink to support the dynamic analysis. (Although CAD data for the 2 DOF robot is available, we did not take it into account for our illustration purposes.) SysML is a general-purpose, block-based modeling language, well suited to formally capture the specifications of a CES in the early design phase. The SysML model captures the robot’s structure, internal element connectivity, connectivity with other units, and behavior. To predict the position control error during the pick-and-place operation, we developed a dynamic control model that predicts the position of the end point of the robot under a control strategy. Further analysis models may also be involved, such as predicting the parts processed per minute or a token flow simulation to analyze a production process plan. The error between the controlled and the desired position is bounded by a constraint captured in the SysML model.

As a result of the overlapping stakeholder concerns, semantic overlaps between models exist. Overlaps appear in the form of semantic relationships between models. For example, the dimensions of the robot arms, the mass and inertia of each arm, the sensor resolution, and the actuator power are specified in the SysML model and are also used in the Simulink model to build a control system for the robot. These duplicate parameters are thus semantically equivalent to each other. Changing these attributes within one model (for example, SysML) should propagate to others (for example, Simulink), or inconsistencies in design will exist. Such equivalence relationships can be found across many models developed to support the robot’s design.

Data Transformations
In preparation for use in our visual analytics tool, we apply three data transformation steps to the data extracted from the RDF triple store. First, we use a base ontology to represent distributed multi-formalism data using a common minimalistic set of vocabulary. Within this base ontology, we use a new class of type Element to refer to any heterogeneous element, such as a SysML or a Simulink Block. Elements relate to other elements through one or more relationships: contain, reference, inheritance, connectedTo, equivalentTo, and satisfy. Because the primary aim of our visual analytics tool is to visualize instance data in heterogeneous models, the second data transformation step involves removing class definitions and OSLC resource shape definitions from the dataset. RDF represents data in subject-predicate-object format, where predicates also take the role of subjects when they are described. This means that, in our node-link visualizations, not all the subject and object types can be considered node types (because they include predicates as well). Similarly, not all the predicate types can be considered link types. The third data transformation step thus involves identifying subjects that are nodes and predicates that are edges. We classify all subjects that are of type Element as nodes; we classify all predicates of the following types as edges: contain, reference, connectedTo, inheritance, satisfy, and equivalentTo.
Visual Computing Challenges

The visualization shown in the user interface is a force-directed network visualization of the entire dataset (see Figure 3). Simulink and SysML nodes are presented in blue and red, respectively. Edges are color-differentiated based on relationship type. Using this representation, a systems engineer can gain several important insights of the relationship within and between Simulink and SysML models. One clear observation is that some elements in the SysML model are standalone and not linked with the rest of the model. This means that some elements are defined but not used currently because these parts of the system are not specified. Another direct observation is that there are many more relations between the Simulink elements than between the SysML elements and also that no relations exist between the two models (no overlaps captured).

With no overlaps formally captured between models, manual coordination is necessary, or inconsistencies might occur. The user can use this network to explore where the potential overlaps are, or they can have the Simulink and SysML modelers identify the overlaps so as to capture them formally.

Expanding on the issue of model overlap, consider a systems engineer who wants to explore and formally capture overlaps between different CES design models. A typical way to do this is by looking at both Simulink and SysML models at a granular level. Filtering by root nodes and their relations using the filter panel, the user modifies the visualization and now sees only root nodes and their connections highlighted, while others are faded into the background to provide the overall model context (see Figure 5). Using this filtered visualization, the systems engineer identifies that LowerArm in SysML

Scenarios

Figure 5. Force-directed network visualization of CES data with node filter root applied. Using this filtered visualization, the user can create equivalence relations between semantically overlapping classes, properties, and relationships.
is referred to as Link1 in Simulink, and UpperArm in SysML is referred to as Link2 in Simulink. The engineer then moves to a more fine-grained level and identifies that the property LinkLength for Elements LowerArm and UpperArm in SysML are duplicated as parameters for L1 and L2 in Simulink. By supporting multiple interaction modalities, such a visualization enables the user to create equivalence relations between semantically overlapping classes, properties, and relationships.

Different visual representations can provide alternative and comparative insights into CES design and support different stakeholder needs. Consider two different stakeholders: a product manager who is interested in how the CES design is organized and its component models distributed; and a systems engineer who wants to understand how a specific design block is connected to other blocks. To accommodate the analysis needs of these users, we developed two different visualizations from the same dataset: a treemap for the product manager (see Figure 6a) and an adjacency matrix for the systems engineer (see Figure 6b). The treemap allows the product manager to understand the overall organization of the CES design while identifying key design blocks. The adjacency matrix representation, on the other hand, allows the systems engineer to explore how the design block are interconnected and the type of connection. Figure 6b, for instance, shows a zoomed-in perspective of the RobotStructure design block in the overall context. The system engineer identifies that connEnd1 is connected to connEnd2; connEnd1 is contained in RobotController and connEnd2 is contained in RobotMechanicalAssembly. The color coding of the cells in the matrix depicts the relationship types, in this case including contain and connectedTo connections.

The previous examples illustrate three different principles of visualization: identifying the stakeholders, determining their concerns, and making a decision about the visualization type that suits the stakeholders. Based on these principles, a view is generated to address the concerns of the stakeholders. With appropriate interaction dynamics, the knowledge gained through exploratory tasks could be formally captured, enabling decision makers to make well-informed decisions.

**Evaluation**

We performed a qualitative, value-driven evaluation with prototypical users to examine four key criteria:
Visual Computing Challenges

- Does the tool help reduce the time to perform routine analysis tasks?
- Does the tool provide insights (confirmatory/novel)?
- Does the tool provide the essence of the underlying data?
- Does it give the user confidence about the data?

Users responses were strikingly positive and suggest, at least anecdotally, that our visual analytics tool is able to support users performing routine CES design analysis tasks, provide insights into the interdependent structure of the models, provided the essence of the data, and gave users confidence in understanding the models. Users also commented on the high usability of our tool suggesting that both visual representations and interaction techniques were appropriate. We fully acknowledge that these results are not generalizable and that a large-scale evaluation of the visual analytics tool is necessary. We are planning to conduct a controlled laboratory experiment and field study to explore these aspects further.

The design and production of CES requires analysis of massive amounts of detailed information, where dealing with multiformalism data available from distributed, disparate, and heterogeneous system modeling sources is a major challenge. We posit that visual analytics is a particularly valuable tool that can help stakeholders interactively explore, discover, and make sense of the underlying data. Our proposed tool takes advantage of two key features to move beyond the capability currently available in common MBSE tools: interoperability using a graph-based representation of models and the capability to generate a view dynamically. We use multiple vocabularies with varying semantic precision to describe the heterogeneous data present in the model repository. We also use a base vocabulary to describe the common port-based network abstraction between different models supporting the CES design. In addition, generating modeling views dynamically and as per the needs of different users rectifies the problems faced with predefined diagrams prevalent in MBSE tools.

Our preliminary results with developing relevant visual analytics tools suggest many opportunities for future work. We envision that future versions of our tool will exhibit three additional capabilities:

- visualizing specifications using semantically rich graphical syntax similar to SysML but augmented with interaction dynamics;
- graphical representations of analysis results, such as temporal simulation results, trade space visualizations, animation of simulation outcomes, and visualization of uncertainty; and
- visualization intended to discover global patterns that are beyond the view of a single stakeholder.

Our aim is to identify the kinds of semantic relationships that can exist in the data from heterogeneous system modeling sources so that we can develop/associate semantically rich graphical syntax to these relationships. We will build upon current visualization capabilities in available system engineering tools and add dynamic browsing, focusing, and zooming capabilities through our visual analytics tool. For example, the part-whole relationship is visualized as an internal block diagram in SysML, which adequately describes how different parts connect to make a whole. However, as the number of parts increase, the internal block diagram becomes cluttered, resulting in reduced readability and understanding. The cluttering can be managed by providing a zoom capability to spot the item of interest, moving the focus to this item to show the internal elements, for example, through a contextual menu. In this way, the focus can be switched between multiple part-whole relationships or from one type of relationship to the other. The contextual menu provides an opportunity to bring up multiple visual representations for an object based on user selection—for example, with a right click or hover-over interaction, a user could select between a 3D representation of the robot, a functional decomposition, or a state chart describing the robot’s behavior.

Acknowledgments

This research was supported by Boeing Research and Technology, the Model-Based Systems Engineering Center, and the Tennenbaum Institute at the Georgia Institute of Technology. We thank Sebastian Herzig, Axel Reichwein, and Timothy Major for their contributions in building the necessary data infrastructure and visual analytic tool to carry out the illustrative case study presented in this article.

References


**Rahul C. Basole** is an associate professor in the School of Interactive Computing, associate director of the Tennenbaum Institute, and a faculty member in the GVU Center at the Georgia Institute of Technology. He is also a visiting scholar in the Human-Sciences and Technologies Advanced Research Institute at Stanford University. His research interests include computational enterprise science, information visualization, and strategic decision support. Basole has a PhD in industrial and systems engineering from the Georgia Institute of Technology. Contact him at basole@gatech.edu.

**Ahsan Qamar** is a postdoctoral fellow at the Model-Based Systems Engineering Center, G.W. Woodruff School of Mechanical Engineering, at the Georgia Institute of Technology. His research interests include model-based systems engineering, combining aspects of decision theory, information technology, simulation, and systems theory to support the design of large-scale, complex engineered systems. Qamar has a PhD in electrical and computer engineering from Carnegie Mellon University. Contact him at ahsan.qamar@gatech.edu.

**Hyunwoo Park** is a Tennenbaum fellow and PhD candidate in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests include technological innovation, platform strategy, computational enterprise science, and visual analytics. Park has an MS in information management and systems from the University of California, Berkeley. Contact him at hwpark@gatech.edu.

**Christiaan J.J. Paredis** is professor of mechanical engineering in the G.W. Woodruff School of Mechanical Engineering and in the H.M. Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology, where he is also a Woodruff faculty fellow and director of the Model-Based System Engineering Center. His research interests include model-based systems engineering, combining aspects of decision theory, information technology, simulation, and systems theory to support the design of large-scale, complex engineered systems. Paredis has a PhD in electrical and computer engineering from Carnegie Mellon University. Contact him at chris.paredis@me.gatech.edu.

**Leon F. McGinnis** is a professor emeritus at the H.M. Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology and a registered professional engineer in the state of Georgia. His research interests include fundamental representation issues in discrete event logistics systems, performance assessment models, and the development of integrated computational tools. McGinnis has a PhD in industrial engineering from North Carolina State University. Contact him at leon.mcginnis@gatech.edu.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.