IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2019, accepted April 27, 2019, date of publication May 9, 2019, date of current version May 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915917

Model-Based Systems Engineering for Aircraft Design With Dynamic Landing Constraints Using Object-Process Methodology

LINWEN LI[1,2,3], NATALI LEVI SOSKIN[2], AHMAD JBARA[2,4], MOTI KARPEL[3], AND DOV DORI[2,5], (Fellow, IEEE)
[1]Shanghai Aircraft Design and Research Institute, Shanghai 201210, China
[2]Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 3200003, Israel
[3]Faculty of Aerospace Engineering, Technion–Israel Institute of Technology, Haifa 3200003, Israel
[4]Computer Science and Engineering Department, University of Connecticut, Storrs, CT 06269, USA
[5]Systems Design and Management, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Corresponding author: Linwen Li (lilinwen@comac.cc)

This work was supported by the National High-End Equipment Talent Development Plan of China.

ABSTRACT We present a model-based systems engineering (MBSE) framework using object-process methodology (OPM, ISO 19450) for civil transport aircraft design with dynamic landing constraints. The framework integrates the aircraft system development life cycle processes into a holistic MBSE model, incorporating into it the specific aircraft dynamic loads model-based design (MBD) domain. Using this framework, we also perform model-based requirements analysis, validation, and verification. The model parameter set is governed by a unified format used for both the MBSE and MBD domains. The resulting model enables aircraft design processes to start being formalized at the early conceptual design stage and carry on to detailed design, adding value to system lifecycle processes by integrating and streamlining the MBSE and MBD domains into a more holistic framework with seamless transition between these two major stages.

INDEX TERMS Model-based systems engineering (MBSE), model-based design (MBD), civil transport aircraft design, dynamic landing constraints, object-process methodology (OPM) ISO 19450.

I. INTRODUCTION

The predominant tasks of an aircraft are not only to fly at peak performance to transport airborne cargo, but also to spend a good part of its life on the ground. According to current airline requirements, an aircraft should sustain up to 90,000 cycles during its lifetime, where a cycle includes take-off, cruise and landing. Statistics show that accidents prior to or directly after the take-off and landing contribute more than 50% to the overall accident figures [1]. During aircraft landing, the loads are transmitted to the airframe through tires, wheels, and landing gear structures. It is therefore of paramount importance to determine the dynamic landing loads accurately during the design phase. This reduces the components' weight, enabling the aircraft to operate more efficiently.

One of the main goals in developing a new aircraft is to balance the structural weight and its integrity to achieve performance and economic benefits. Structures of modern aircrafts become increasingly flexible. The main reasons for this are slender wings and fuselages that frequently arise from the stretching of existing aircraft and the use of new, light-weight structures and materials, which influence the vibrational properties of the fuselage and the wings [2]. At the same time, for landing gears, the tire's pneumatic physics and the absorber's stiffness and damping nature are the biggest nonlinear characteristic contributions to the aircraft landing dynamics, as various dynamic elements must be taken into consideration. To understand the aircraft landing dynamics and determine the aircraft design under these constraints, a sophisticated, holistic design methodology, based on a formal, accurate model, is necessary.

Dynamic landing loads applied on the airframe during the landing phase are of great concern in the design of modern aircrafts, because the loads result in not only material fatigue; they also affect the aircraft structural integrity [1]–[4]. Many engineering methods and tools, such as the multi-body system

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhai Zhang.

modeled using a Model-Based Design (MBD) approach, aim to ensure the aircraft flight safety and operational economics. These methods and tools may obtain the structural response under relevant loads and improve the structure capability to withstand extreme loads by a specific design solution [5]. However, from a whole-system perspective, an aircraft is an extremely complex human-made product that comprises many systems, components and parts, which together perform the function of passenger and cargo transporting. The biggest challenge in aircraft design is the emergence of complexity that surrounds the numerous components' and systems' design and integration [6]. Many of these complexities have resulted in significant technical obstacles for the aircraft designer, making it difficult to come up with a good solution. Tough challenges stem from the multidisciplinary nature of the problem, raising the question of how disparate disciplines can come together in the aircraft design cycle [7]. To overcome these challenges, over the last decades, aircraft manufactures have been first to adopt a Systems Engineering (SE) approach for efficient design synthesis in the multidisciplinary flight domain.

Traditional SE processes are document-based: the system is described in a set of text documents, such as requirement documents, system design specifications, and interface control documents. The information contained in these documents is difficult to maintain, synchronize, and assess in terms of its quality, such as correctness, completeness, and consistency. Model-Based Systems Engineering (MBSE), which relies on a formal conceptual modeling language and methodology, such as OPM – Object-Process Methodology [8], ISO 19450: 2015, was developed to formalize the SE processes by using an evolving model throughout the system development lifecycle. In MBSE, the conceptual model is the source of reference and authority—the equivalent of the blueprints of machine drawings used in the classical engineering of the 1950s through 1970s.

This paper proposes a MBSE framework using OPM to implement civil transport aircraft design with dynamic landing constraints, into which the dynamic elements of the aircraft body and its landing gear are integrated. The V-model of SE processes, described in the sequel, provides a holistic MBSE architecture, in which detailed MBD models of the aircraft interact with the governing, ''big-picture'' OPM model to exchange the configured parameters during the aircraft components implementation and integration processes.

The rest of the paper is organized as follows: Section II presents our methodological foundations and principles as they are applied to construct the OPM V-model. Section III illustrates case studies of implementing the MBSE-MBD framework in aircraft design, specifically the design of an aircraft landing gear with dynamic landing constraints using OPM. In Section IV, we present our aircraft landing gear design case study. Section V illustrates design realization procedures with emphasis on design verification, while Section VI concludes with a discussion, implications, benefits, and future work.

II. MBSE METHODOLOGY

A. *OBJECT-PROCESS METHODOLOGY*

Object-Process Methodology, OPM, ISO 19450: 2015, is founded on a minimal ontology of stateful objects as things that exists, or can exist, physically or informatically, and processes as things that transform objects by creating or consuming them, or by changing their state. OPM is a holistic approach to specifying systems, which integrates the structural, functional and behavioral aspects of a system into a single, unified model with built-in refinement-abstraction mechanism, expressed bi-modally in formal yet intuitive graphics and equivalent text in a subset of a natural language.

As illustrated in Table 1, OPM elements are things and links. An OPM thing can be an object (rectangle), or a process (ellipse). Objects can be stateful, i.e., have states. Processes and objects exhibit two properties (inherent attributes): Essence, which can be informatical or physical, and Affiliation, which can be systemic or environmental. Processes and objects may consist of lower-level processes and objects. Links express graphically various relations between these elements. Structural links support modeling of static system aspects, and can be defined between two objects, two processes, and in some cases between an object and a process. Procedural links express dynamic relations, transformations of objects by processes, flow of control, with event and conditions to process execution.

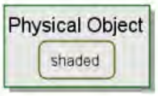
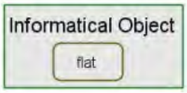
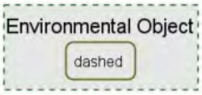
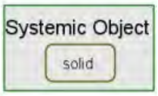
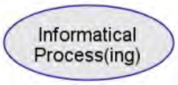
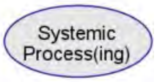
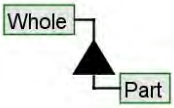
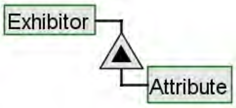
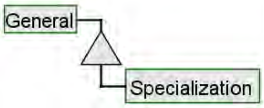
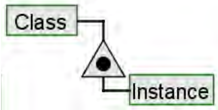
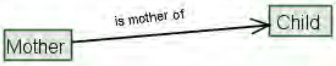
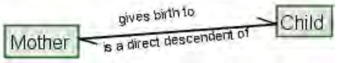
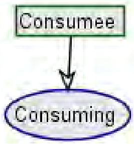
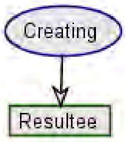
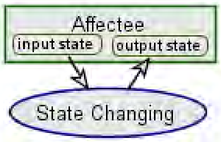
The graphical modality is a set of self-similar, notation-consistent, hierarchically organized, interconnected, and cross-validated Object-Process Diagrams (OPDs) – the only kind of OPM diagram. An OPD contains things – stateful objects and processes – connected with several kinds of links. The textual modality, Object-Process Language (OPL), is a structured textual specification in a subset of a natural language – English or any other language. Each graphical OPD construct is also expressed by a semantically equivalent textual OPL sentence, which is instantly created in response to the modeler's graphic input. Table 1 shows the main OPM elements (things and relations) in their graphical (OPD) and textual (OPL) modalities. These OPD symbols and OPL sentences are used in the OPM models in the sequel.

B. *V-MODEL OPM FORMALIZATION*

1) V-MODEL ABSTRACT FORMALIZATION

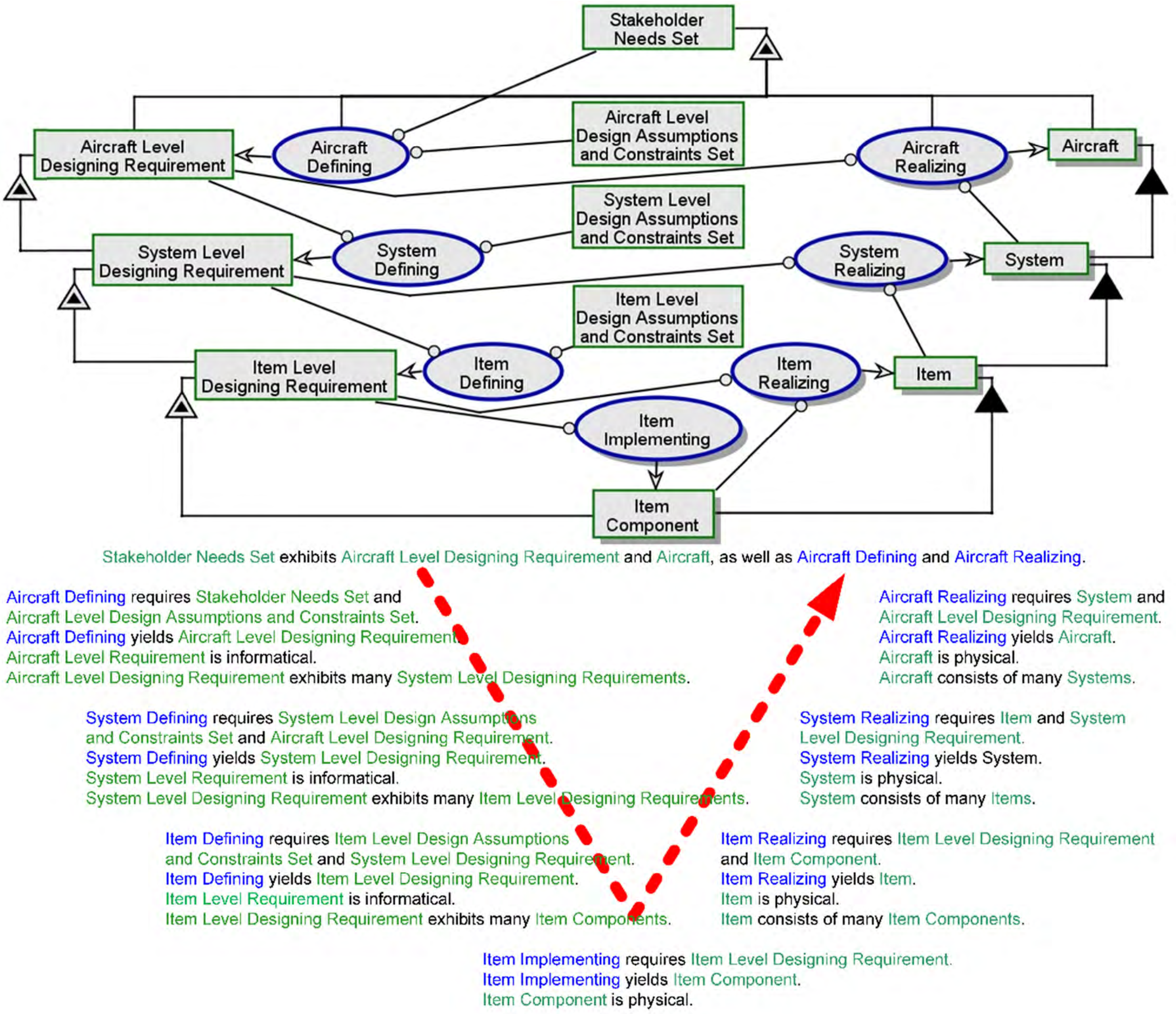
The V-model is an accepted SE method that has been introduced to aircraft and its systems development for improved SE focus, particularly during the concept and development stages [9]. The V-model highlights the requirements for continuous validation with the stakeholders, the verification during the requirements development and implementation, and the importance of continuous risk and opportunity assessment. The V-model provides a useful illustration of the SE activities during the aircraft and systems development lifecycle stages. Fig. 1 expresses the V-model of an aircraft development lifecycle in OPM. The SE activities are formalized by OPM things processes, denoted graphically by ellipses,

TABLE 1. Object-Process Methodology elements outline.

OPM Things: Stateful Object and Process				
Modality	Stateful Object			
Graphical (OPD)				
Textual (OPL)	Physical Object is shaded.	Informational Object is flat.	Environmental Object is dashed.	Systemic Object is solid.
Modality	Process			
Graphical (OPD)				
Textual (OPL)	Physical Process(ing) is physical.		Environmental Process(ing) is environmental.	
OPM Links: Structural Links and Procedural Links				
Modality	Fundamental Structural Links			
	aggregation-participation link	exhibition-characterization link	generalization-specialization link	classification-instantiation link
Graphical (OPD)				
Textual (OPL)	Whole consists of Part.	Exhibitor exhibits Attribute.	Specialization is a General.	Instance is instance of a Class.
Modality	Tagged Structural Links			
	unidirectional tagged link	bidirectional tagged link	Reciprocal tagged link	
Graphical (OPD)				
Textual (OPL)	Mother is mother of Child.	Mother gives birth to Child. Child is a direct descendent of Mother.	Child and Mother are family.	
Modality	Procedural Enabling Links			
	agent link	instrument link		
Graphical (OPD)				
Textual (OPL)	Agent handles Processing.	Processing requires Instrument.		
Modality	Procedural Transforming Links			
	consumption link	result link	effect link	in-out link pair
Graphical (OPD)				
Textual (OPL)	Consuming consumes Consumee.	Creating yields Resultee.	Affecting affects Affectee.	State Changing changes Affectee from input state to output state.

and objects—by the boxes. Following the defining processes and the resulting informatical objects while descending on the left leg of the V-model, we go from **Aircraft Defining**—the process of defining the entire aircraft, which results in the informatical (not shaded) object **Aircraft Level Designing Requirement**, through **System Defining** to **Item Defining**.

At the bottom of the V-model is the process **Item Implementing**, which requires the object **Item Level Designing Requirement** and yields the physical (shaded) object **Item Component**. Following the realizing processes and the resulting physical objects while also ascending on the right leg of the V-model, we go from **Item Realizing**, which results



Stakeholder Needs Set exhibits Aircraft Level Designing Requirement and Aircraft, as well as Aircraft Defining and Aircraft Realizing.

Aircraft Defining requires Stakeholder Needs Set and Aircraft Level Design Assumptions and Constraints Set.
Aircraft Defining yields Aircraft Level Designing Requirement.
Aircraft Level Requirement is informatical.
Aircraft Level Designing Requirement exhibits many System Level Designing Requirements.

Aircraft Realizing requires System and Aircraft Level Designing Requirement.
Aircraft Realizing yields Aircraft.
Aircraft is physical.
Aircraft consists of many Systems.

System Defining requires System Level Design Assumptions and Constraints Set and Aircraft Level Designing Requirement.
System Defining yields System Level Designing Requirement.
System Level Requirement is informatical.
System Level Designing Requirement exhibits many Item Level Designing Requirements.

System Realizing requires Item and System Level Designing Requirement.
System Realizing yields System.
System is physical.
System consists of many Items.

Item Defining requires Item Level Design Assumptions and Constraints Set and System Level Designing Requirement.
Item Defining yields Item Level Designing Requirement.
Item Level Requirement is informatical.
Item Level Designing Requirement exhibits many Item Components.

Item Realizing requires Item Level Designing Requirement and Item Component.
Item Realizing yields Item.
Item is physical.
Item consists of many Item Components.

Item Implementing requires Item Level Designing Requirement.
Item Implementing yields Item Component.
Item Component is physical.

FIGURE 1. Systems Engineering V-model transformed to an OPM model: OPD (top) and its corresponding OPL (bottom).

in the object **Item**, all the way to the entire physical **Aircraft**. Each realizing stage requires the set of constituents from the previous stage and the requirement of the object realized at the current stage.

As illustrated in the OPD in Fig. 1, all the aircraft level processes and objects are linked to the top object, **Stakeholder Needs Set**, with an OPM exhibition-characterization structural link, indicating that all the development efforts are performed to satisfy the needs of the stakeholders, which is the basic goal of the aircraft design. The OPL sentences equivalent to the graphic input are created or modified instantly by the OPM modeling software (OPCAT[1] or OPCloud)[2] as the modeler edits the OPD. The OPL sentences at the bottom of Fig. 1 express in detail the model's static and dynamic aspects expressed graphically by the OPD at the top of Fig. 1. The OPL sentences accurately indicate the information and materials flow during the product or system evolution and maturation, as expressed by the V-model.

2) REFINING THE V-MODEL BY IN-ZOOMING

Process in-zooming is an OPM refinement mechanism that can be applied iteratively to any process. We apply in-zooming to processes in the OPD in Fig. 1 to elaborate on each process and expose more detailed perspectives. The in-zoomed processes are presented below in separate OPDs, each being part of the big OPM V-model. In each OPD, one of the V-model processes exposes a set of subprocesses, and in parallel, the aggregate objects are unfolded to expose a set of specialized or constituent objects. The refined OPDs for all the in-zoomed processes have a similar structure regarding the aircraft, system and item levels.

---

[1] OPCAT can be downloaded from esml.iem.technion.ac.il/opcat-installation/and used freely.

[2] See https://www.opcloud.tech/

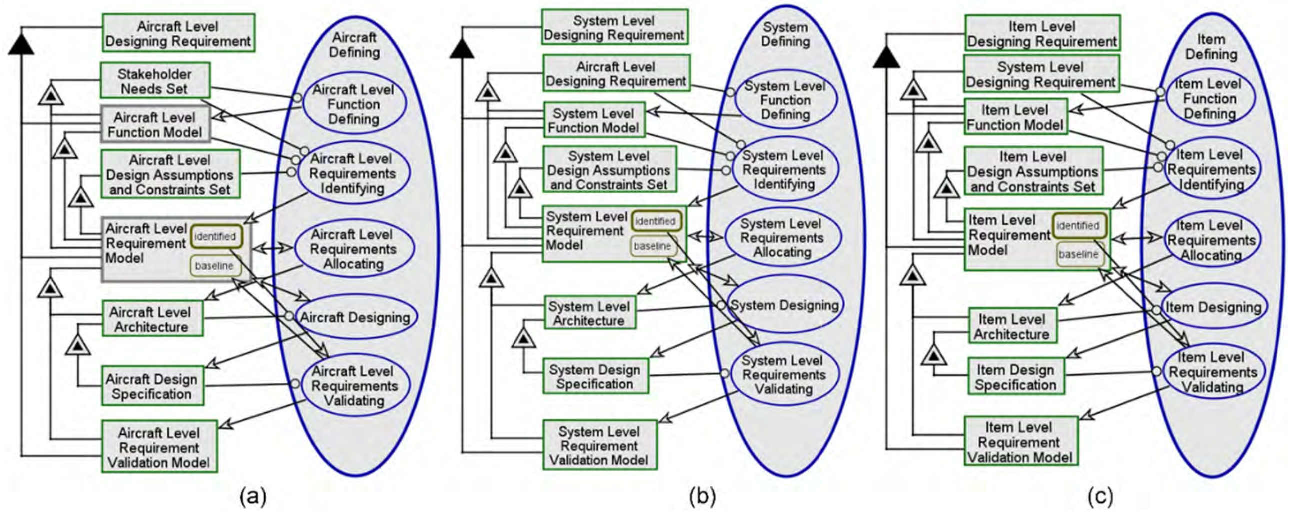
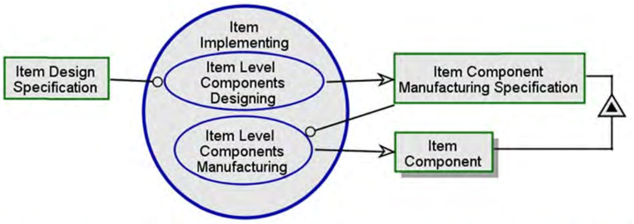


FIGURE 2. The left leg of V-model refined: (a) in-zoomed Aircraft Defining (b) in-zoomed System Defining (c) in-zoomed Item Defining.

For example, as shown in Fig. 2(a), at the highest level, the **Aircraft Defining** process is zoomed into five subprocesses: **Aircraft Level Function Defining, Aircraft Level Requirements Identifying**, **Aircraft Level Requirements Allocating, Aircraft Designing**, and **Aircraft Level Requirements Validating. Aircraft Level Function Model** is the output object of the subprocess **Aircraft Level Function Defining**, and it uses the object **Stakeholder Needs Set**. The subprocess **Aircraft Level Requirements Identifying** elicits the identified **Aircraft Level Requirement Model** from the **Stakeholder Needs Set, Aircraft Level Function Model**, as well as **Aircraft Level Design Assumptions and Constraints Set**. The identified **Aircraft Level Requirement Model** triggers the subprocess **Aircraft Level Requirements Allocating** to allocate the requirements to the **Aircraft Level Architecture**, and it generates the corresponding **Aircraft Design Specification** from the subprocess **Aircraft Designing**. The effect links indicate the iterative activities of the subprocesses to gradually refine the **Aircraft Requirement Model** during the **Aircraft Defining** process. The **Aircraft Design Specification** is the instrument—the enabling object of the subprocess **Aircraft Level Requirements Validating**, in which the aircraft level requirements are validated, changing the state of **Aircraft Level Requirement Model** from **identified** to **baseline**. Finally, the validation data is recorded in the corresponding object **Aircraft Level Requirement Validation Model**. The baseline **Aircraft Level Requirement** is an instrument for the lower level defining process. The in-zoomed OPDs at the system and item levels, shown in Fig. 2(b) and Fig. 2(c), respectively, have similar structure and behavior at more refined levels.

As shown in Fig. 3, the **Item Implementing** process refines the informatical object **Item Design Specification** to **Item Component Manufacturing Specification** by the **Item Level Components Designing** subprocess. The information flow is used to transform the specifications into real physical entities, in which the **Item Components** are produced by the **Item Level Components Manufacturing** subprocess. This implies that the **Item Implementing** process is the connection between the information and materials flows during the aircraft development lifecycle. The corresponding OPL sentences are automatically generated to express this OPD's structural relationships and procedural behavior during the information and material flows.



Item Implementing consists of Item Level Components Designing and Item Level Components Manufacturing.
Item Implementing zooms into Item Level Components Designing and Item Level Components Manufacturing.
Item Level Components Designing requires Item Design Specification.
Item Level Components Designing yields Item Component Manufacturing Specification.
Item Level Components Manufacturing requires Item Component Manufacturing Specification.
Item Level Components Manufacturing yields Item Component.
Item Component is physical.
Item Component Manufacturing Specification exhibits Item Component.

FIGURE 3. The V-model Item Implementing process in-zoomed framework.

The realizing processes integrate the end product and verify that it has been implemented correctly at the appropriate level. As illustrated in Fig. 4, the in-zoomed V-model right side presents the detailed realizing processes perspective. For example, at the lowest level in Fig. 4(a), the **Item Realizing** process is zoomed into three sequential subprocesses: **Item Level Components Assembling**, **Item Level Requirements Verifying**, and **Item Certifying**. The **Item Level Components Assembling** requires physical **Item Component** and informatical **Item Design Specification** to integrate the physical **Item**, creating it at its **assembled** state. The subprocess **Item Level Requirements Verifying** verifies the associate **Item Level Requirement Model**, in which the item level

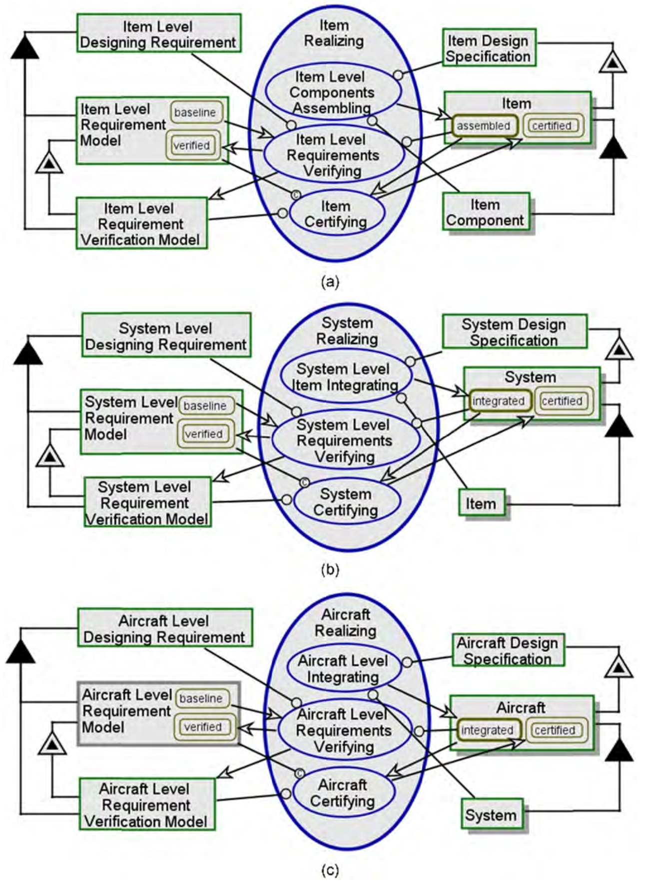
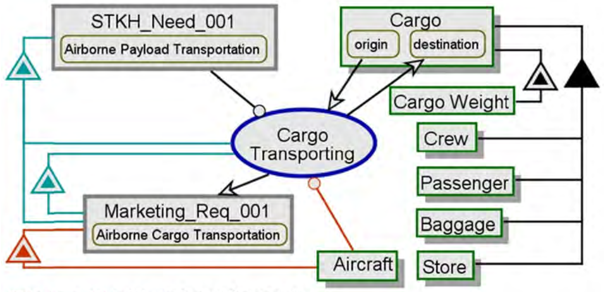


FIGURE 4. **The right leg of V-model refined: (a) in-zoomed Aircraft Realizing (b) in-zoomed System Realizing (c) in-zoomed Item Realizing.**

requirements are verified, changing the state of **Item Level Requirement Model** from **baseline** to **verified**, the verification data is recorded in the corresponding object **Item Level Requirement Verification Model**. When all the item level requirements are verified, the resulting **Item Level Requirement Verification Model** is the instrument that enables performing the subprocess **Item Certifying** to certify the **Item**, changing its state from **assembled** to **certified**. At this stage, all the items are ready to be used for the higher-level system integration. A similar pattern is applied to higher levels of the realizing process, until the entire aircraft is integrated, verified, and finally certified for delivery and deployment. The refined OPDs are illustrated in Fig. 4(b) and Fig. 4(c) for the system and aircraft levels, respectively.

By formalizing the traditional SE V-model in an OPM conceptual modeling framework and refining it into detailed in-zoomed OPDs using OPM's refinement-abstraction mechanisms (in-zooming and unfolding), the relationships between the objects and processes are clearly illustrated by the static structural exhibition-characterization link of OPM, providing full traceability from requirements to realization. This is important for managing SE procedures during the aircraft development lifecycle. For example, the structural relationships between the objects are modeled by OPM's aggregation-participation link to indicate whole-part information and the bill-of-materials structure. Specifically, the object **Aircraft Level Designing Requirement** aggregates and can be traced to the objects **Aircraft Function Model**, **Aircraft Level Requirement Model, Aircraft Level Requirement Validation Model**, as well as **Aircraft Level Requirement Verification Model**.



STKH_Need_001 is Airborne Payload Transportation.
Cargo Transporting Requires STKH_Need_001.
Cargo Transporting yields Marketing_Req_001.
Marketing_Req_001 is Airborne Cargo Transportation.
Marketing_Req_001 exhibits Aircraft.
Cargo Transporting requires Aircraft.
Aircraft is physical.
Cargo Transporting changes Cargo from origin to destination.

Function Defining
Requirements Identifying
Requirements Allocating

STKH_Need_001 exhibits Marketing_Req_001, as well as Cargo Transporting.
Cargo Transporting exhibits Marketing_Req_001.

Traceability

Cargo is physical.
Cargo can be origin or destination.
Cargo exhibits Cargo Weight.
Cargo consists of Passenger, Baggage, Store, and Crew.
Passenger is physical.
Baggage is physical.
Store is physical.
Crew is physical.

Configuration management

FIGURE 5. **Top level design definition analysis with formalized OPM V-model.**

III. DESIGN DEFINITION

A. TOP LEVEL DESIGN DEFINITION

From a SE perspective, the driver of a new aircraft development is the market needs from potential customers. As shown in Fig. 5, the market needs are presented in an object **STKH_Need_001**[3] to express the goal of the needed aircraft—its top-level function—is **Airborne Payload Transportation**. Following the **Function Defining** process of the OPM V-model, the fundamental top-level function process **Cargo Transporting** results from **STKH_Need_001**. The top-level beneficiary is the physical object **Cargo**, which is transported from **origin** to **destination** by the **Cargo Transporting** process. In fact, we could better model the human interested in transporting the cargo as the real beneficiary. The **Requirements Identifying** process elicits a requirement set expressed in the model as **Marketing_Req_001** to clarify the functional aspects translated from informal stakeholder needs to a formal requirement statement. The OPL sentence "**Cargo Transporting** requires **Aircraft**." implies that **Marketing_Req_001** is allocated to the **Aircraft** to constraint its design for achieving the **Cargo Transporting** function.

The allocation relationship is illustrated in the OPD in Fig. 5 by the red exhibition-characterization link (line

---

[3] STKH stands for Stakeholder

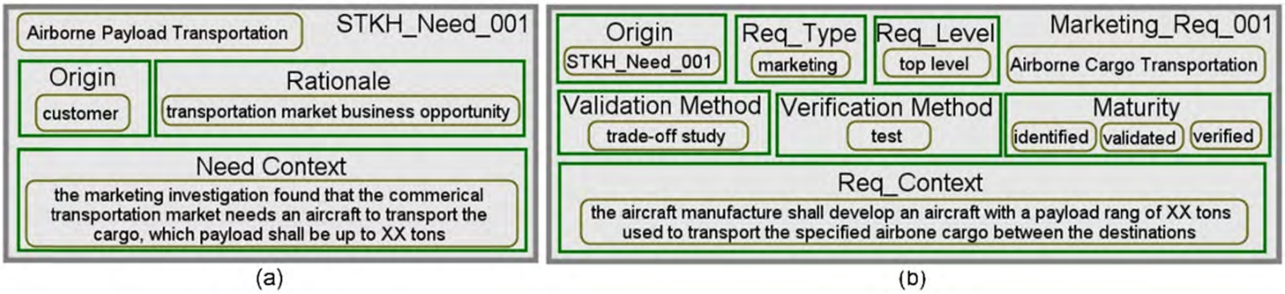


FIGURE 6. Stakeholder need and requirement objects in-zoomed: (a) stakeholder need model (b) marketing requirement model.

with black-in-white triangle along it). After the processes **Function Defining, Requirements Identifying**, and **Requirements Allocating** are performed, the resulting objects are modeled and linked using OPM exhibition-characterization links with cyan color, as expressed also by the following corresponding OPL sentences: **STKH_Need_001** exhibits **Marketing_Req_001**, as well as **Cargo Transporting**, and **Cargo Transporting** exhibits **Marketing_Req_001**. Now, the top-level design definition is complete, and it captures the original function and requirement. **Cargo** can now be defined as a configuration item set with attributes and aggregated parts for the aircraft level design definition stage.

OPM provides the ability to trace objects in a formal way. For example, Fig. 6 illustrates that the objects **STKH_Need_001** and **Marketing_Req_001** from Fig. 5 are further in-zoomed into a more detailed need model and requirement model, respectively. As shown in Fig. 6(a), **STKH_Need_001** consists of three lower-level objects: **Origin, Rationale**, and **Need Context**. The unique identifier of the whole model is the primary object name **STKH_Need_001**, and it will be used as reference to all traceability citations during the corresponding SE procedures and activities. Its state clearly indicates that the model's core objective and delivered value is **Airborne Payload Transportation**. The value of the object **Origin, customer**, indicates that the source of the need is the customer**. Rationale** has the value **transportation market business opportunity**, indicating why the aircraft development is necessary. Finally, **Need Context** can be used for recording as its value specific content, such as the detailed need statement. Similarly, the object **Marketing_Req_001** can also be in-zoomed into a detailed requirement model that consists of seven-part objects, such as the one illustrated in Fig. 6(b).

B. AIRCRAFT LEVEL DESIGN DEFINITION

To achieve the goal of the top-level function **Cargo Transporting**, we zoom into this process, as illustrated in the OPD in Fig. 7. Following the **Function Defining** process of the OPM V-model in Fig. 2(a), **Cargo Transporting** zooms into three sequential aircraft-level processes: **Aircraft Take-off Operating, Aircraft Cruising**, and **Aircraft Landing**. First, the **Cargo** is loaded on **Aircraft** (not modeled). Then, the process **Aircraft Take-off Operating** changes the loaded **Aircraft** from state **on ground** to **in air**, and the **Cargo** departs from the origin airport. **Aircraft Cruising** keeps the **Aircraft** efficiently flying from **origin** to **destination**. The process **Aircraft Landing** changes the **Aircraft** from state **in air** back to **on ground**, and the **Cargo** finally arrives at the destination airport.

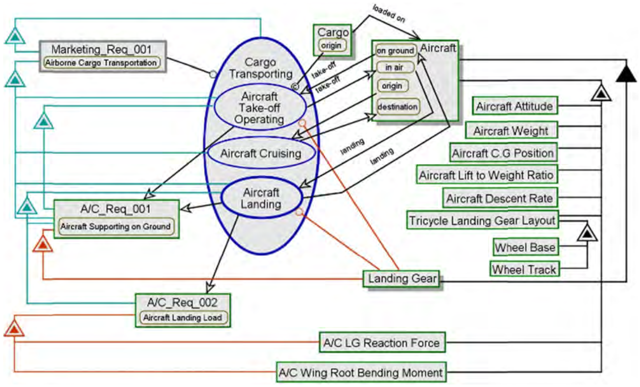


FIGURE 7. Cargo Transporting process in-zoomed.

Our focus in this paper is aircraft design issues that account for dynamic load constraints during landing, so we focus on and zoom into the process **Aircraft Landing**. A related aircraft level functional requirement **A/C_Req_001–Aircraft Supporting on Ground** is identified. **Aircraft Landing** can be directly traced to **Marketing_Req_001**. The requirement is allocated to the (physical and systemic) object **Landing Gear** to achieve the **Aircraft Landing** function. To constraint the function and the aircraft design, an aircraft level derived requirement **A/C_Req_002–Aircraft Landing Load** is also identified and allocated to objects **A/C LG**[4] **Reaction Force** and **A/C Wing Root Bending Moment**, for which **Aircraft Landing Load** is specified.

An important output of the aircraft level design definition process is the **Aircraft Level Architecture** and **Aircraft Level Design Specification**, which meet the aircraft design with dynamic landing constraints. The OPL sentences that

---

[4] LG is landing gear.

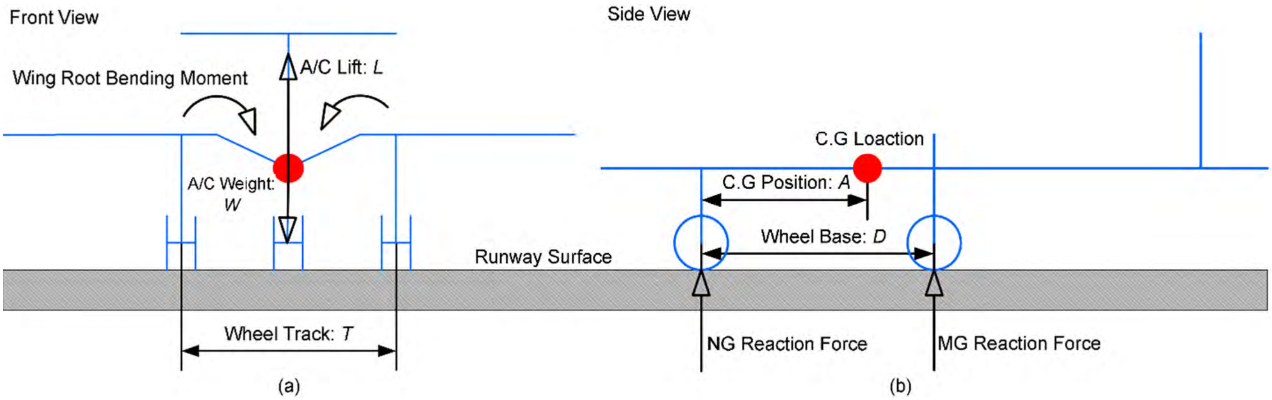


FIGURE 8. Illustrations of some of the aircraft level design parameters modeled in Fig. 7.

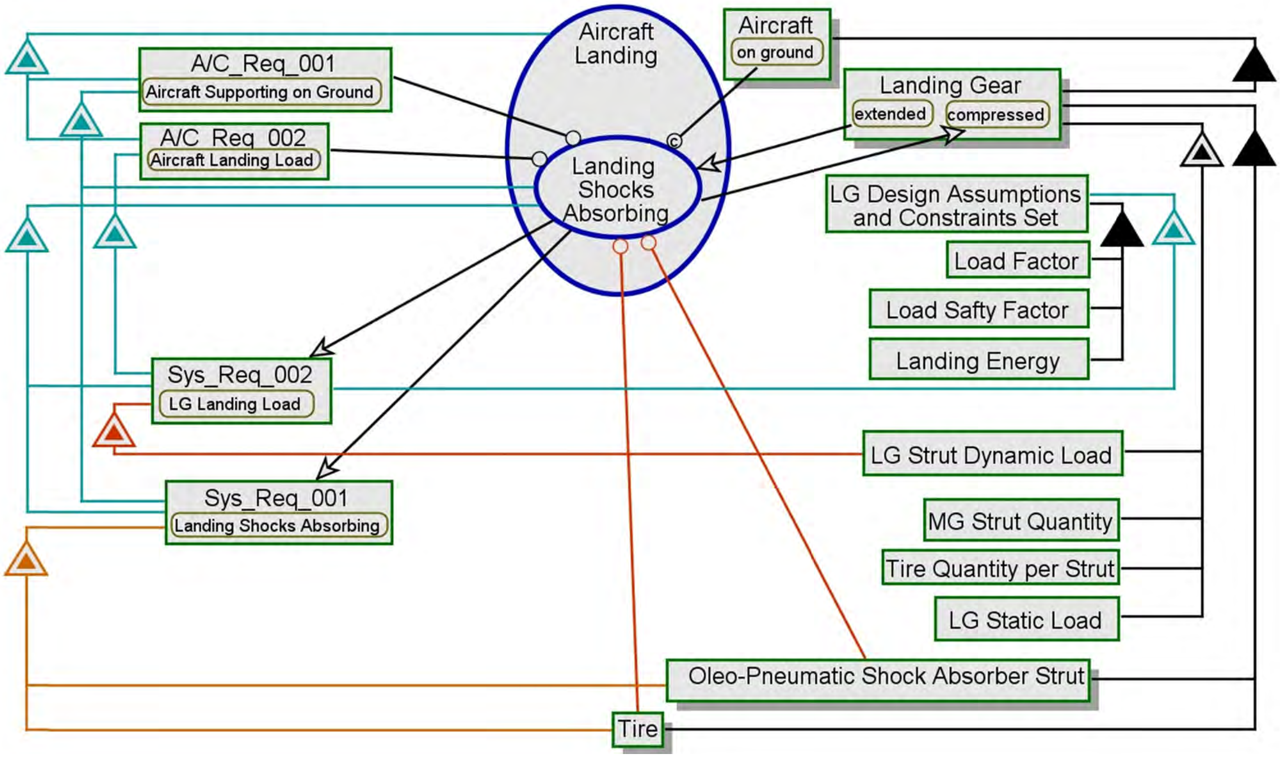


FIGURE 9. System level design definition analysis with formalized OPM V-model.

were automatically generated from the corresponding OPD can be used to produce these outputs. As illustrated in Fig. 7, the structural relation between the **Aircraft** and its **Landing Gear** is modeled by OPM's aggregation-participation link, resulting the **Aircraft Level Architecture** specification "**Aircraft** consists of **Landing Gear**." Finally, the **Aircraft Level Design Specification** is constructed from the **Aircraft** attributes (objects that characterize **Aircraft**), expressed in the following OPL sentences: (1) **Aircraft** exhibits **Aircraft Weight**, **Aircraft C.G Position, Aircraft Lift to Weight Ratio, Aircraft Attitude, Tricycle Landing Gear Layout, A/C LG Reaction Force,** and **A/C Wing Root Bending Moment**. (2) **Tricycle Landing Gear Layout** exhibits **Wheel Track** and **Wheel Base**. The physical meaning of some of these attribute objects, which are design parameters, is illustrated in Fig. 8.

C. SYSTEM LEVEL DESIGN DEFINITION

As shown in Fig. 9, from the function defining perspective, when **Aircraft** state changed from **in air** to **on ground**, namely the **Aircraft** performs a touch-down, a system level functional process **Landing Shocks Absorbing** is defined from the high-level requirements **A/C_Req_001** and **A/C_Req_002**, identified during the aircraft level design definition stage described earlier. The function **Landing Shocks Absorbing** is used to allow the aircraft landing energy to be absorbed with the state of **Landing Gear** changing from **extended** to **compressed** during the functional process **Aircraft Landing** operations. The instruments required to achieve this function are **Oleo-Pneumatic Shock Absorber Strut and Tire**.

Following the requirements identifying process formalized in the OPM V-model, a system-level functional requirement **Sys_Req_001** is identified, which can be directly traced to the aircraft level functional requirement **A/C_Req_001**. In it, the objects **Oleo-Pneumatic Shock Absorber Strut** and **Tire** are allocated as parts of **Landing Gear** to facilitate the **Landing Shocks Absorbing** process. In addition, a system-level performance requirement, **Sys_Req_002**, is identified from **LG Design Assumptions and Constraints Set**, as it significantly affects the **LG Landing Load** properties. The **Landing Gear** architecture and design is specified in the OPD in Fig. 9, and the design process can now proceed down to the item level, described next.

D. ITEM-LEVEL DESIGN DEFINITION

To manage the complexity of the system to be designed, an abstract-to-details strategy is applied to the item level design stage due the large number of item-level entities. As shown in the OPD in Fig. 10, when **Landing Gear** is compressed to achieve the system level process **Landing Shocks Absorbing**, two item-level functional processes, **Energy Dissipating** and **Energy Storing**, are defined from the system-level requirements **Sys_Req_001** and **Sys_Req_002**. Three effect links indicate the abstract relationships between these processes and **Landing Gear** items. For instance, the effect link between the process **Energy Storing** and the object **Tire** implies that the function is allocated to the tire, and the tire's state will be changed as the process occurs.

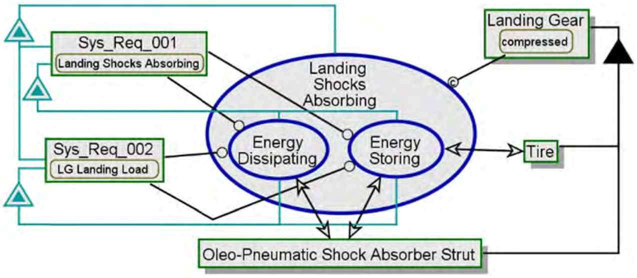


FIGURE 10. **Landing Shock Absorbing process in-zoomed.**

Fig. 11 illustrates the item-level detailed design of two processes. As shown in Fig. 11(a), when **Landing Gear** is compressed during aircraft touch-down, **Energy Storing** is carried out by changing the state of **Tire** from **inflated** to **compressed**, and the state of **Oleo-Pneumatic Shock Absorber Strut** from **extended** to **compressed**, in order to store the energy in the tires and absorber struts. Following the **Requirements Identifying** process, an item level functional requirement **Item_Req_001–Landing Energy Storage** is identified from the defined function, which traces to system level requirement **Sys_Req_001**. The identified requirement is allocated to four item-level component objects: **Main Strut Pneumatic Spring**, **Nose Strut Pneumatic Spring**, **Main Tire**, and **Nose Tire**. Thus, the final technical solution for landing energy storage and absorption is ultimately implemented at the lowest level design definition stage. Meanwhile, two item-level performance requirements, **Item_Req_002–Tire Dynamic Load** and **Item_Req_003–Strut Spring Load**, are identified from system level requirement **Sys_Req_002**, as well as from **Tire** and **Strut Design Assumptions and Constraints Set**, where the goal of the item components designing is formally specified.

A separate model of the process **Energy Dissipating** is similarly illustrated in Fig. 11(b). By combining the separate OPDs in Fig. 11 into a comprehensive perspective, the strut's architecture can be expressed as **Oleo-Pneumatic Shock Absorber Strut**, which consists of **Main Strut Pneumatic Spring**, **Main Strut Hydraulic Damper**, **Nose Strut Pneumatic Spring**, and **Nose Strut Hydraulic Damper**. The tire's architecture is similarly expressed in the OPL sentence "**Tire** consists of **Nose Tire and Main Tire**." **Item Level Design Specification** can be constructed from the design assumptions and constraints set and all the attributes of the struts and tires expressed in Fig. 11. At this stage we move to mathematics-based, quantitative modeling of the details of the design implementation. Our future objective is to enable seamless transition to this stage using OPM, which is currently being extended with computational capabilities. At this stage, though, from this point on, we use only mathematical modeling, as our integrated conceptual-computational OPM modeling framework, implemented in OPCloud, while already operational, is not yet mature enough to accommodate this challenge, and we are working to make this seamless transition possible. Once we are done with the mathematical modeling, we return to the OPM model and the requirements it expresses to close the loop.

IV. DESIGN IMPLEMENTATION

A. DYNAMIC ELEMENTS INTRODUCTION

As introduced in the design definition stage, the dynamic loads during aircraft landing need to be absorbed by the landing gear struts and tires, thereby reducing the shocks transmitted to the airframe. A schematic diagram of a shock absorbing mechanism that operates during the aircraft landing process is described in Fig. 12. The mechanism is based on the principle of compressing the gas in the struts and tires during the landing gear dropping process to store the energy. Meanwhile, the stored energy is dissipated by the hydraulic oil being forced through a small orifice. These dynamics elements of a pneumatic spring and a hydraulic damping are modeled in detail next.

B. TIRE DESIGN AND MODELING

The tire vertical force F_{ti} on the tire-runway interface due to polytropic compression of gas inside the tire can be

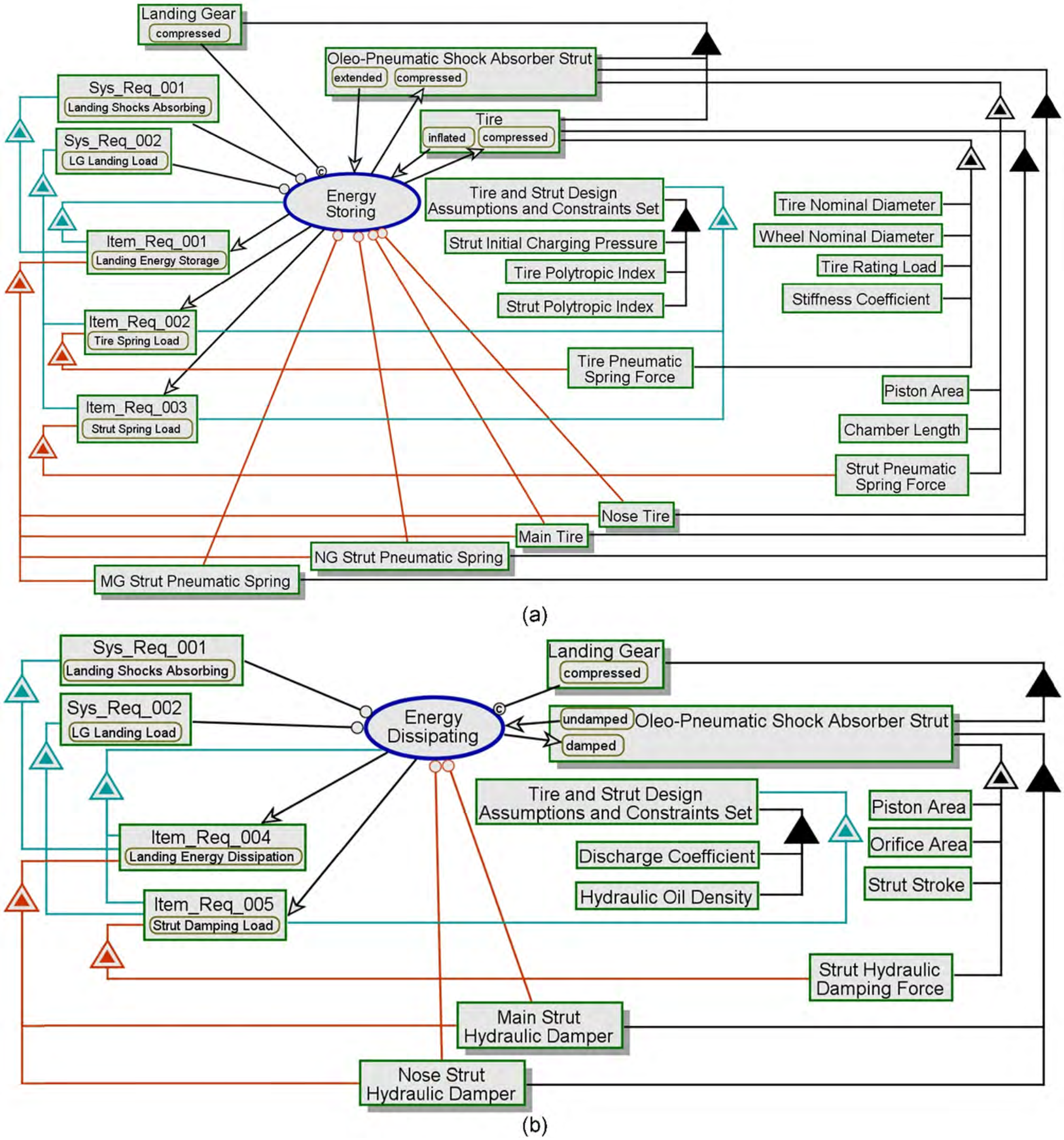


FIGURE 11. Item-level detail design of two processes: (a) energy storing (b) energy dissipating.

expressed by

$$F_{ti} = K_{ti} \times \left(\frac{x_{ti}}{d_{ti}}\right)^{r} \tag{1}$$

where the parameters K_{ti} and d_{ti} are the stiffness coefficient and nominal diameter of the tire respectively, which are determined to fit the given load deflection characteristics for the tire being used. The constant r is a polytropic index for which 1 is generally used. The variable x_{ti} is the vertical deflection of the tire that represents the change in the tire diameter between the loaded and unloaded conditions.

To configure the tire pneumatic spring parameters, represented in Eq. (1), a tire selection process needs to be made based on its rated load condition. To accomplish the tire selection, the static loads on each gear strut need to be determined. The static load per strut for a given gear is

$$\begin{cases} F_{s,N} = \dfrac{D-A}{D}W \\ F_{s,M} = \dfrac{1}{N}\dfrac{A}{D}W \end{cases} \tag{2}$$

where W is a specified aircraft weight corresponding to a critical aircraft center of gravity (C.G.) position, and N is main gear strut quantity, for which 2 is used due to the fact that a tricycle landing gear was selected for the aircraft design. The other parameters are defined in Fig. 8.

The tire rating load can be estimated as

$$\begin{cases} F_{Sti,N} = \dfrac{1.07 \times F_{S,N}}{2} \le F_{rating,N} \\ F_{Sti,M} = \dfrac{1.07 \times F_{S,M}}{2} \le F_{rating,M} \end{cases} \tag{3}$$

where the tire quantities are 2 for both nose and main gears, and 1.07 is a safety factor for tire's rating loads design. Based on this criterion, the tire selection for the designed aircraft can be made.

Daugherty [10] points out that a tire which statically supports the rated load at its rated pressure will cause a deflection to approximately 24 ∼ 35% of its available deflection before it bottoms out on the wheel. Therefore, an equation between the tire rating load and static deflection may be found from Eq. (1) as

$$F_{rating} = K_{ti} \times \frac{S_{ti}}{d_{ti}} \tag{4}$$

Assuming that the unloaded distance between the wheel flange and the radius of the tire is taken as $(d_{ti} - d_w)/2$, and the tire deflection at the rated load is approximately 1/3 of this unloaded distance. Therefor the tire's rated deflection becomes

$$S_{ti} = \frac{1}{3}\left(\frac{d_{ti} - d_w}{2}\right) \tag{5}$$

where the diameters of the tire and wheel, d_{ti} and d_w, are available from the tire manufacturer when a given tire is selected.

By substituting $\mathrm{S_{ti}}$ into Eq. (4), the tire stiffness coefficient can be found as

$$K_{ti} = F_{rating}\left[\frac{1}{6}\left(1 - \frac{d_w}{d_{ti}}\right)\right]^{-1} \tag{6}$$

C. STRUT PNEUMATIC SPRING DESIGN AND MODELING

For Oleo-pneumatic spring modeling, assuming a gas compression in the strut is an adiabatic process, the gas pressure and volume can be represented by the polytropic equation as

$$PV^k = P_1 V_1^k \tag{7}$$

where P and V are the gas pressure and volume at any compression conditions. The gas initial charging pressure and volume are P_1 and V_1, respectively. The exponent k is 1.1 for adiabatic process.

As shown in Fig. 12(a) and Fig. 12(b), the length of a gas chamber in initial and loaded conditions is l and l−x, respectively. For a given piston area A_h, the gas initial volume and loaded volume can be denoted as $V_1 = A_h l$, and $V = A_h(l - x)$. By substituting the V_1 and V into Eq. (7), the gas spring force $F_{\text{spring}} = A_h P$ becomes

$$F_{spring} = A_h P_1 \left(\frac{1}{1 - \frac{x}{l}}\right)^{1.1} \tag{8}$$

where x is the vertical distance of the Oleo-pneumatic strut that represents the change in the strut traveling distance between the loaded and unloaded conditions.

Fig. 12(c) depicts a landing gear at rest under the static loading due to the weight of the aircraft. For this load condition, the pressure of the compressed gas in a strut, P_2, is assumed to be about 1500 psi. The ratio of the static load pressure to the pressure in the fully extended strut is usually set at about 4, as shown in Fig. 12 (a). Thus, in the fully extended position, the gas pressure, P_1, would drop to about 375 psi. In the most stressing touchdown case, shown in Fig. 12(d), the internal force supported by the strut is nF_S and therefore the pressure $P_3 = nP_2$, for which the load factor $n = 3$ is generally assumed. Thus, in the fully compressed position, the gas pressure, P_3, would increase to about 4500 psi.

With an airplane at rest, the force acting on the piston in the strut is $F_S = P_2 A_h$. Therefore, the required piston area of the strut is

$$A_h = \frac{F_S}{P_2} \tag{9}$$

where the maximum static gear strut load, F_S, under aircraft having maximum gross weight and critical C.G. combination is calculated from Eq. (2) for the main gear and the nose gear, respectively.

When an aircraft is under the most stressing touchdown, as shown in Fig. 12 (d), the landing gear strut travel is the strut stroke S. The volume occupied by the gas when the landing gear is fully compressed is $V_3 = A_h(l\text{-}S)$, and the ratio of the fully compressed load pressure to the pressure in the fully extended strut is $P_3/P_1 = 12$. By substituting these parameters into Eq. (7), the strut gas chamber length can be found as

$$l = \frac{S}{1 - \sqrt[1.1]{\frac{1}{12}}}. \tag{10}$$

where the calculated strut maximum gas chamber length is generally increased by adding 0.75∼1 inch margin to withstand overstressed situations for aircraft operations.

During an aircraft touchdown, the total of the kinetic and potential energy should be equal to the energy to be absorbed by the Oleo-pneumatic struts and tires with their vertical stroke. The energy equilibrium can be expressed by

$$\frac{1}{2}\frac{W}{g}v^2 + (W - L)(S + S_{ti}) = \eta n S W + \eta_{ti} n S_{ti} W \tag{11}$$

where the parameters, W, v, and L are the aircraft weight, sink rate, and lift force, respectively. n is the load factor,

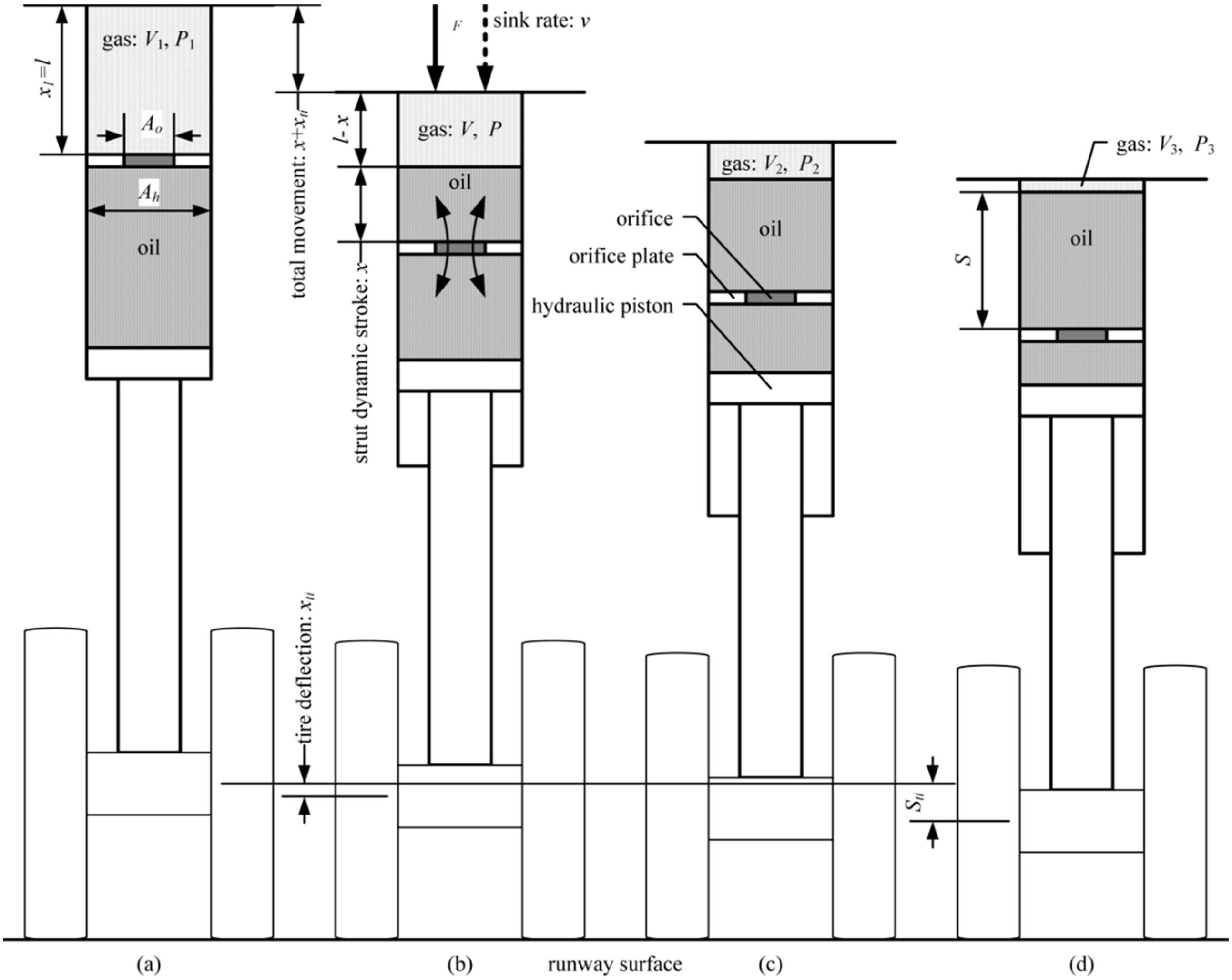


FIGURE 12. A schematic diagram of shock absorbing during aircraft landing: (a) fully extended (b) dynamically loaded (c) statically compressed (d) fully compressed.

while η and η_{ti} are the process efficiency for Oleo-pneumatic struts and tires, respectively, and g is the gravity constant. By introducing the lift-to-weight ratio L/W, the required strut stroke is obtained by

$$S = \frac{v^2}{2g\left(\eta n - 1 + \frac{L}{W}\right)} - S_{ti}\left(\frac{\eta_{ti} n - 1 + \frac{L}{W}}{\eta n - 1 + \frac{L}{W}}\right) \tag{12}$$

where the aircraft sink rate, v, is to be 12ft/s, and typical values for efficiencies are $\eta = 0.8$ and $\eta_{ti} = 0.5$. The load factor is assumed as $n = 3$, and the ratio between the lift and weight may be assumed as $L/W = 1$ for a commercial transport aircraft. The tire deflection S_{ti} can be estimated from Eq. (5).

Now the parameters configured in Eq. (8) for the Oleo-pneumatic strut spring force are readily available from Eqs. (9), (10) and (12) and the assumptions. The landing gear strut size, including the hydraulic piston area A_h, strut gas chamber length l, and initial charging pressure P_1, are also determined.

D. STRUT HYDRAULIC DAMPING DESIGN AND MODELING

For Oleo-pneumatic strut damping force modeling, an orifice flow and pressure equation may be expressed by

$$A_h \dot{x} = C_d A_o \sqrt{\frac{2\Delta P}{\rho_h}} \tag{13}$$

where the parameter A_o is the orifice area, ρ_h is the hydraulic oil density, and C_d is the discharge coefficient. ΔP is the differential pressure measured across the orifice. Therefore, the hydraulic damping force, $F_{\text{damping}} = A_h \Delta P$, can be found from Eq. (13) as

$$F_{damping} = -sgn\left(\dot{x}\right) \frac{\rho_h A_h^3}{2\left(C_d A_o\right)^2} \dot{x}^2 \tag{14}$$

where the hydraulic fluid used in landing gear shock struts is a mineral-based fluid with a specific density ρ_h of about 880kg/m^3. For a rounded entry orifice, the discharge coefficient may reasonably be taken as $C_d = 0.9$. The area of

TABLE 2. Aircraft configuration parameters.

<table>
<tr><th colspan="10">Aircraft Configuration Item Parameters</th></tr>
<tr><td rowspan="3">Aircraft Attitude
level landing</td><td colspan="3">Aircraft Weight (kg)</td><td colspan="2">Aircraft C.G. Position (m)</td><td colspan="2">Aircraft Lift to Weight Ratio</td><td colspan="2">Tricycle Landing Gear Layout(m)</td></tr>
<tr><td>MRW</td><td>MTOW</td><td>MLW</td><td>Fwd C.G.</td><td>Aft C.G.</td><td>landing</td><td>drop test</td><td>Wheel Base</td><td>Wheel Track</td></tr>
<tr><td>W =71000</td><td>W =70500</td><td>W =61500</td><td>$A = 5$</td><td>$A = 5.4$</td><td>$L/W = 1$</td><td>$L/W = 0$</td><td>$D = 6$</td><td>$T = 5$</td></tr>
<tr><th colspan="10">Landing Gear Configuration Item Parameters</th></tr>
<tr><td colspan="4">LG Design Assumptions and Constraints Set</td><td colspan="6">LG Designing Parameters</td></tr>
<tr><td rowspan="2">Load Factor</td><td rowspan="2">Safety Factor</td><td rowspan="2">MG Strut Quantity</td><td rowspan="2">Tire Quantity per Strut</td><td colspan="6">LG Static Load (lb)</td></tr>
<tr><td colspan="3">NG Maximum Static Load
Nose Tire Maximum Static Load</td><td colspan="3">MG Maximum Static Load per Strut
Main Tire Maximum Static Load</td></tr>
<tr><td rowspan="2">$n = 3$</td><td rowspan="2">1.07</td><td rowspan="2">$N = 2$</td><td rowspan="2">2</td><td rowspan="2">$F_{S,N} = 26088$
$F_{Sti,N} = 13957$</td><td colspan="2">Load Condition</td><td rowspan="2">$F_{S,M} = 70438$
$F_{Sti,M} = 37684$</td><td colspan="2">Load Condition</td></tr>
<tr><td>Fwd C.G.</td><td>W = MRW</td><td>Aft C.G.</td><td>W = MRW</td></tr>
</table>

Tire and Oleo-Pneumatic Shock Absorber Strut Configuration Item Parameters

Tire and Strut Design Assumptions and Constraints Set

Tire Polytropic Index	Strut Polytropic Index	Discharge Coefficient	Hydraulic Oil Density (kg/m^3)	Initial Charging Pressure (psi)
$r = 1$	$k = 1.1$	$C_d = 0.9$	$\rho_h = 880$	$P_1 = 375$

Tire Designing Parameters

Tire Rating Load (lb)		Tire Nominal Diameter (inch)		Wheel Nominal Diameter (inch)		Stiffness Coefficient (N)	
nose tire	main tire	nose tire	main tire	nose tire	main tire	nose tire	main tire
$F_{rating} = 14340$	$F_{rating} = 40600$	$d_{ti} = 30$	$d_{ti} = 43.5$	$d_w = 15$	$d_w = 20$	$K_{ti} = 7.649*10^5$	$K_{ti} = 2.004*10^6$

Oleo-Pneumatic Shock Absorber Strut Designing Parameters

Piston Area (m^2)		Chamber Length (m)		Orifice Area (m^2)		Strut Stroke (m)	
nose gear	main gear	nose gear	main gear	nose gear	main gear	nose gear	main gear
$A_h = 1.121*10^{-2}$	$A_h = 3.028*10^{-2}$	$l = 0.4574$	$l = 0.4323$	$A_o = 9.882*10^{-5}$	$A_o = 2.128*10^{-4}$	$S = 0.3869$	$S = 0.3644$

the piston, A_h, was determined from Eq. (9). The negative sign function, $-sgn(\dot{x})$, indicates that the damping force is always against the direction of the strut movement. This is fundamental to dissipating the shock energy during the aircraft landing process.

To calculate the parameter of the orifice area A_o represented in Eq. (14), Cook and Milwitzky [11] modeled a solution for the normalized maximum stroke as a function of the normalized vertical sink rate, which can be approximated by

$$S_{n,max} = 1.74 v_n^{0.58} \tag{15}$$

where the normalized maximum stroke of the landing gear piston, $S_{n,max}$, and the normalized vertical sink rate of the aircraft, v_n, are defined as

$$\begin{cases} S_{n,max} = \dfrac{\rho_h A_h^3}{2\left(C_d A_o\right)^2 \cos\varphi} \dfrac{g}{F_s} S \\ v_n = \dfrac{\rho_h A_h^3}{2\left(C_d A_o\right)^2 \cos\varphi} \dfrac{g}{F_s} \sqrt{\dfrac{S_{ti}}{g}} v \end{cases} \tag{16}$$

The quantity φ is the angle that the landing gear strut makes with the vertical axis. For a commercial transport aircraft, φ is small enough to permit the approximation $\cos\varphi \sim 1$. The quantity F_S is the weight carried by the strut, which can be calculated from Eq. (2). Therefore, everything is known, except the orifice area A_o. Substituting for $S_{n,max}$ and v_n in Eq. (15), the required orifice area can be found as

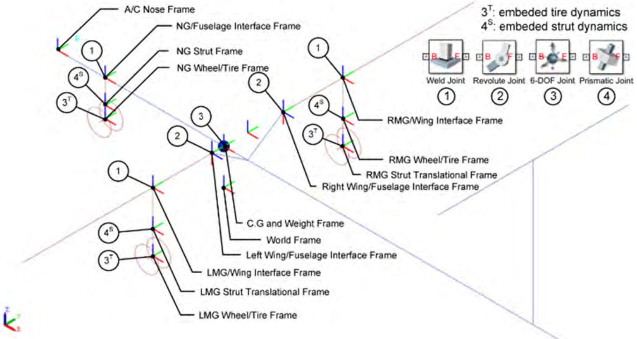


FIGURE 13. GTA multi-body models with dynamic landing elements.

$$A_o = \left[\frac{\left(\dfrac{\rho_h A_h^3 g}{2 C_d^2 F_S} \right)^{0.42} S}{1.74 \left(\sqrt{\dfrac{S_{ti}}{g}} v \right)^{0.58}} \right]^{\frac{1}{2\times 0.42}} \tag{17}$$

E. DYNAMIC ELEMENTS OF THE IMPLEMENTED CONFIGURATION

Table 2 lists the aircraft design parameters produced from the specification of the OPM model at different design levels.

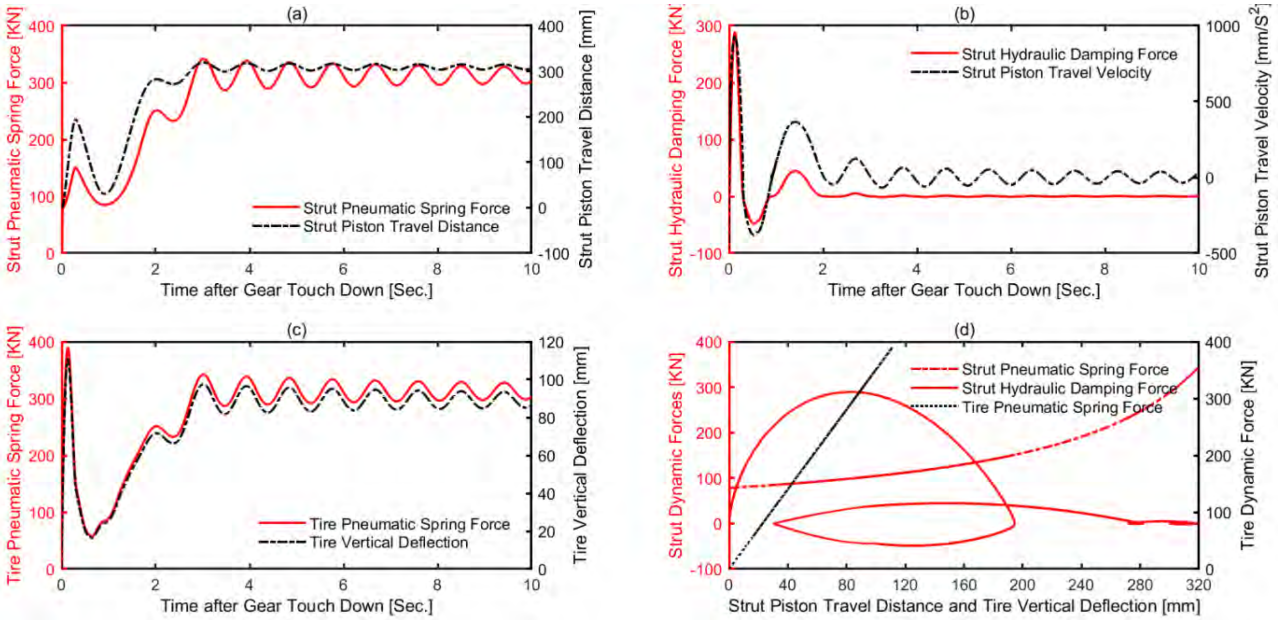


FIGURE 14. Item level design realization results: (a) strut pneumatic spring dynamic force (b) strut hydraulic damping dynamic force (c) tire pneumatic spring dynamic force (d) item components performance characteristic.

Using these parameters, the configuration items of aircraft design with dynamic landing constraints are established.

V. DESIGN REALIZATION

A. AIRCRAFT MULTI-BODY DYNAMIC ELEMENTS

Fig. 13 illustrates a Generic Transportation Aircraft (GTA) multi-body system that is based on the GTA model [12]. To simplify the landing simulation for the purpose of this paper, the model was modified by adding a tricycle landing gear and removing its flexibility, except for rotation springs at the wing roots to obtain the wing bending effects on the landing gear reaction forces. In this model, the aircraft is represented by a simple beam structure, such as a fuselage beam, wings beam, and landing gears with tire and shock absorber. The tricycle landing gear is rigidly attached to the fuselage and wings. The lift force is applied at the C.G. position in the vertical translational direction. The shock absorbers (Oleo-pneumatic strut) of the landing gears are located between the shock tube and the wing interface. Each shock absorber has one degree-of-freedom (DOF) prismatic joint that is embedded into the strut's pneumatic spring and hydraulic damping to model the nonlinear spring and damping contributions to the landing dynamics. Each landing gear includes two wheels and tires assembly, and the pneumatic physics between the tires and runway are modeled as 6-DOF joints that are embedded into the tire's pneumatic spring element, in which the tire reaction force in the vertical translation direction is dynamically applied.

The dynamic elements design utilizes a Model-Based Design (MBD) methodology based on Simscape^{TM} of MATLAB modeling environment, with the design parameters managed in the OPM V-model, as listed in Table 2. The simulation activities are performed as a whole system by integrating the models of GTA and the designed dynamic elements, for which the requirement is to be verified under the specified test case.

B. ITEM LEVEL DESIGN REALIZATION

To verify the requirements defined at the item level design definition stage, a test case at the **Item Level Requirement Verification Model** is established. At this point, we are done with the mathematical modeling, and we refer back to the OPM model and the requirements it expresses to close the loop.

From a functional perspective, the requirement **Item_Req_001** defines the function **Landing Energy Storage**, which reads: *The landing gear apparatus shall provide the energy storage capability to store the vertical kinetic and potential energy of the aircraft during its touch-down.* As shown in Fig. 14(a) and Fig. 14(c), the spring forces produced from the allocated items strut and tire clearly indicate that the corresponding spring force is a function of the strut piston travel distance and tire vertical deflection, respectively. This is because the landing energy is transformed to potential energy of the pneumatic springs when the strut and tire are gradually compressed from zero to the stable compression range. In addition, the requirement **Item_Req_004** defines another function: *the landing gear apparatus shall provide the energy dissipation capability to damp down the oscillation in a stable position with limited cycles during the aircraft landing operation*. Fig. 14(b) illustrates that the strut hydraulic damping force is a function of the strut piston travel velocity. After the aircraft touch-down, the damping force decreased to practically zero within two oscillation cycles,

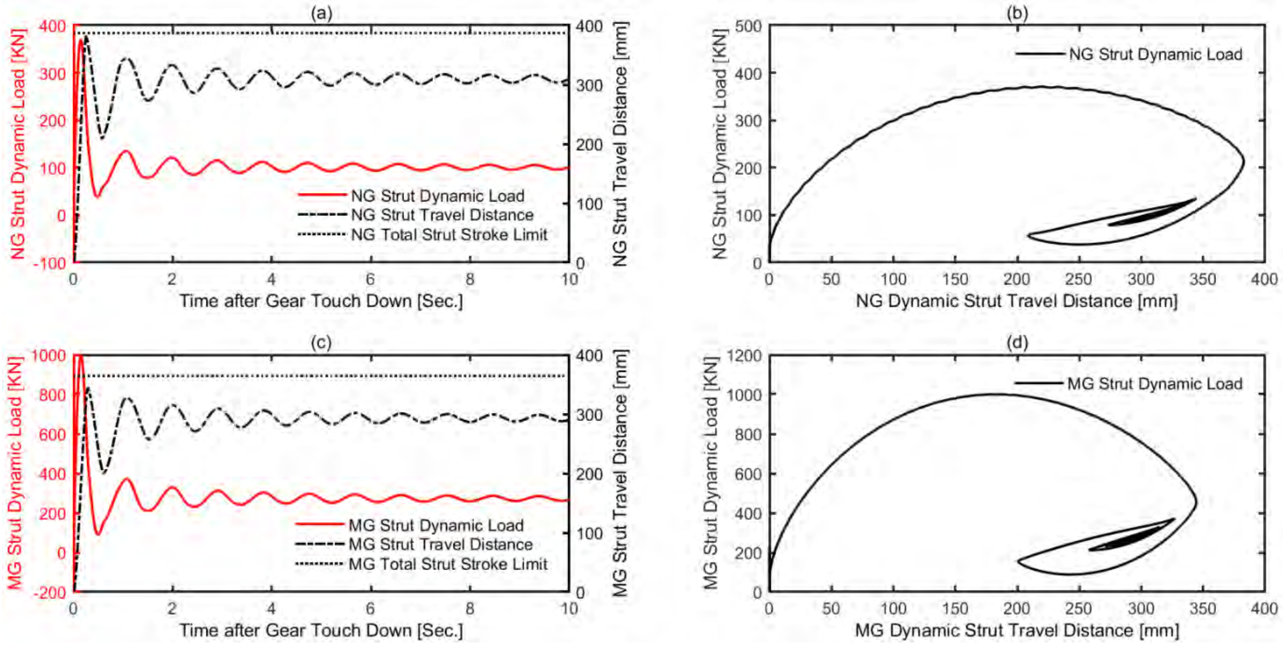


FIGURE 15. System-level design realization results: (a) nose landing gear shocks absorbing characteristic; (b) nose landing gear dynamic force; (c) main landing gear shocks absorbing characteristic; (d) main landing gear dynamic force.

and the oscillation amplitude is also reduced to a smaller range from a higher shock duration. Therefore, these two item-level functional requirements are finally verified by the evidence from these test case simulation results.

The performance requirements specified in item-level requirements **Item_Req_002**, **Item_Req_003**, and **Item_Req_005** also need to be verified with the implemented data. For example, the **Tire Dynamic Load** performance, defined in the requirement **Item_Req_002**, expresses that *the tire shall be able to withstand in the specified landing load conditions*. As shown in Fig. 14(d), the implemented test case simulation shows that the tire load linearly follows the tire's vertical deflection. This is an important result that can be used to determine the tire's extreme position when it is under the specified critical load conditions, which verifies its dynamic load performance. Similarly, the **Strut Spring Load** performance, defined in **Item_Req_002**, can also be verified by fitting the strut pneumatic spring force curve to find out the strut extreme stroke in the specified critical load conditions. **Item_Req_005** defined the performance of **Strut Damping Load** as follows: *the strut damping force shall provide the ability to reduce the loaded oscillation into a stable state within limited number of cycles*. As shown in Fig. 14(d), the implemented strut damping force has a significant hysteresis nature when the strut piston dynamically travels after touch-down. This verifies **Item_Req_005**, as the oscillation is eliminated to zero damping force condition within one damping hysteresis cycle to stabilize the strut in a compressed position, in which the stable force is applied from the strut spring. At this point, the state of all the item-level requirement models changes from **baseline** to **verified**, enabling each item to change its state from **assembled** to **certified**, so the items are ready for system-level design realization, discussed next.

C. SYSTEM LEVEL DESIGN REALIZATION

By integrating the separate struts and tires into a whole designed tricycle landing gear, the system-level design realization procedures are carried out, as we outline in this section. To verify the requirements defined at the system-level design definition stage, a virtual landing gear drop test is performed, based on an integrated multi-body system. The test case initial conditions are taken from the FAR Part 25 regulation §25.723–absorption tests, and the simulation configuration is represented as follows: The aircraft descent velocity is 12 fps, the aircraft weight is set up as the maximum landing weight (MLW), and the aircraft lift to weight Ratio (L/W) is zero. The test case is aggregated into a **System Level Requirement Verification Model** that manages the verification data.

The requirement **Sys_Req_001** defines the function **Landing Shocks Absorbing** in detail, specifying that *the landing gear system shall provide the capability to absorb the landing shocks during the aircraft landing operations*. As shown in Fig. 15(a) and Fig. 15(c), the time-domain data of the test case simulation result illustrate that the shock loads in the landing gear struts can gradually converge to a steady state by combining the contributions of the strut spring and the damping forces. The spatial-domain curves illustrated in Fig. 15(b) and Fig. 15(d) also demonstrate that the landing gear strut dynamic loads can be decreased to the corresponding stable position with its hysteresis properties. Therefore, the requirement **Sys_Req_001** is verified from the test case time-domain

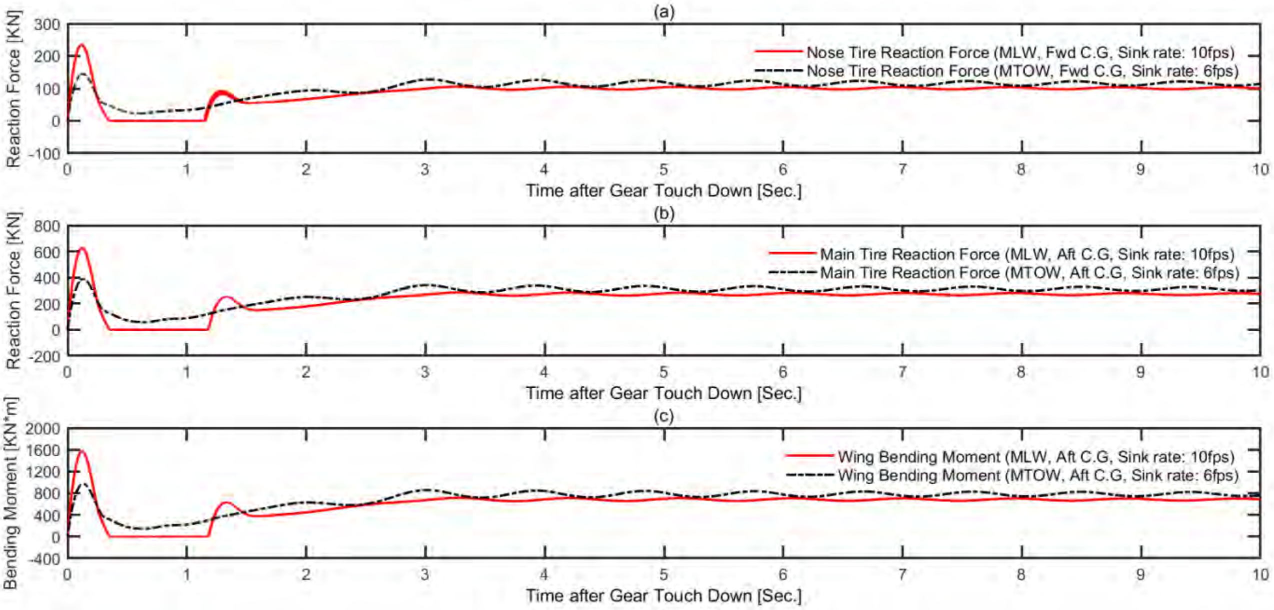


FIGURE 16. **Aircraft level design realization results: (a) nose tire reaction force (b) main tire reaction force (c) wing root bending moment.**

and spatial-domain data, as the intended function is achieved by the designed landing gear.

The requirement **Sys_Req_002** specified that *the landing gear shall be able to withstand when specified critical load conditions are applied*. To demonstrate this performance, the critical load conditions for nose gear and main gear are specified under the Fwd C.G. and Aft C.G. positions, respectively. The allowable limit stroke for the nose gear is denoted in Fig. 15(a) by a straight dashed line. The simulation results clearly indicate that the strut travel distance is always below the allowed stroke value of the piston. This means that no over-travel issues are encountered when the specified critical load conditions are applied to the nose gear. Similarly, the main gear load performance is demonstrated in Fig. 15(c), where the main gear strut travel distance is always below its allowable limit stroke. Therefore, the system level requirements **Sys_Req_001** and **Sys_Req_002** are verified by the virtual drop test simulation results, and they can be used to move on to the aircraft-level design realization, the subject of the next section.

D. AIRCRAFT-LEVEL DESIGN REALIZATION

The requirement **A/C_Req_001** defined that the aircraft shall provide the capability to support itself when it maneuvers on the ground. It is obvious that the allocated aircraft system, **Landing Gear**, whose main function is to support the aircraft on the ground, meets this requirement. Another important issue of aircraft design is determining the aircraft landing loads used for mechanical interface management. In this paper, two test cases are established to figure out the corresponding limit landing loads. These test cases configured the aircraft under level landing attitude, the aircraft lift-to-weight ratio was taken as $L/W = 1$ since the design is for a civil transport aircraft. For one case, the aircraft descent velocity is 10 fps, in which the aircraft weight is MLW. For the other case, the aircraft descent velocity was taken to be 6 fps with maximum take-off weight (MTOW). The test cases were calculated considering only the inertial contribution of the lumped mass as concentrated at the C.G. position, ignoring the inertial forces in the multi-body model due to the elastic response of the wings. The aircraft wings are assumed simply to be a cantilever rigid beam mounted on a root bending spring. The maximum bending moment would occur at the wing-root position due to the landing gear reaction forces acting on the aircraft, ignoring the inertial forces due to the wing response.

The requirement **A/C_Req_002** defined that *the aircraft structure shall be able to withstand the expected limit landing load when the aircraft is under the specified load conditions*. Fig. 16(a) illustrates the nose tire maximum reaction force resulting from the combined landing load conditions of MLW, Fwd C.G., and 10 fps descent velocity during the aircraft level landing process. This trend clearly indicates that the aircraft descent velocity is a major factor of the aircraft landing load contribution, yielding a higher landing energy during the aircraft touchdown process. Similarly, Fig. 16(b) also shows the main tire maximum reaction force resulting from the combined landing load conditions of MLW, Aft C.G. and 10 fps descent velocity during the aircraft level landing process. As shown in Fig. 16(c), the maximum wing-root bending moment is produced from the same landing load condition as the maximum main tire reaction force. This is true because the bending springs modeled in the interfaces between the wings and fuselage, as shown in Fig. 13, are generated in our model by the associated tire reaction force. The above maximum limit landing loads can now be configured into the aircraft level requirement verification model for the corresponding structure design and optimization to comply with the requirement **A/C_Req_002**.

At this point, all the design activities related to our aircraft with dynamic landing constraints, which we focused on as a case-in-point using Model-Based Systems Engineering (MBSE) with Object-Process Methodology (OPM) are completed.

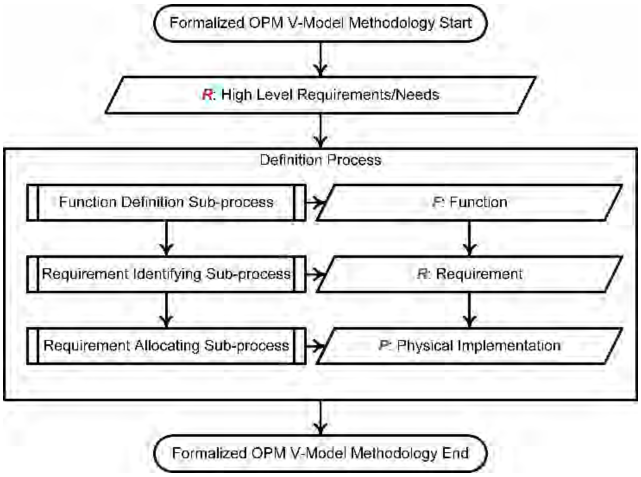


FIGURE 17. A high level "how-to" flowchart schematic for the formalized OPM V-Model methodology.

VI. CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this paper, we have demonstrated a formalized model-based systems engineering (MBSE) approach based on the V-model using OPM. The key flowchart of the methodology was drawn out as Fig. 17, namely as R-F-R-P schematic. We defined and carried out a specific end-to-end aircraft design problem, namely the high-level conceptual design of a civil transport aircraft, based on abstract requirements to transport cargo from origin to destination, and how it gradually percolates to a detailed design of that aircraft's landing gear that meets specific aero-mechanical requirements related to factors such as energy absorption and oscillation damping during the critical touchdown moment while the aircraft is landing. The benefits of the formalized OPM V-model are the following: (1) improved communication among the aircraft stakeholders and across development stages by capitalizing on OPM as a formal yet intuitive modeling language and methodology with its bimodal expression in graphics—the OPD set, and textual—the OPL specification; (2) increased ability to manage the system complexity by OPM's refinement (in-zooming and unfolding) mechanisms to allow the system to be specified hierarchically in a holistic model from high-level abstract user requirements all the way to the physical item-level components needed to meet those requirements; (3) improved system management capability by formalizing the document-based requirements and SE activities in a model that follows a MBSE approach with OPM, and which can be evaluated for consistency, correctness, and completeness.

The main limitation of the current work is the abrupt transition we had to make while switching to the quantitative aspects of the various landing gear parameters, where we left OPM and used traditional mathematical modeling. We then returned to the requirements and high-level aircraft structure and behavior that were modeled in OPM and showed how each requirement is satisfied by the quantitative modeling.

To overcome this limitation, in a near-future work, currently underway, we are augmenting the capabilities of OPM-based modeling by incorporating the quantitative aspects expressed in this paper as an integral part of the OPM model. Using these augmented OPM capabilities, which will become part of the future OPM ISO 19450, we are already able to perform the computations specified in this work within the OPM model. We do this by assigning computational capabilities to OPM processes and numeric or symbolic values to OPM objects. Computations can use a host of predefined, built-in functions or user-programmable ones. Where needed, we can also interface with engineering design software packages, such as MATLAB/Simulink used in this paper. Following this trajectory, we envision OPM with its OPCloud modeling platform as an integrating host of a combined MBSE and MBD ecosystem.

REFERENCES

[1] W. Krüger, I. Besselink, D. Cowling, D. B. Doan, W. Kortüm, and W. Krabacher, "Aircraft landing gear dynamics: Simulation and control," *Int. J. Vehicle Mech. Mobility*, vol. 28, pp. 119–158, Aug. 1997.
[2] P. D. Khapane, "Simulation of asymmetric landing and typical ground maneuvers for large transport aircraft," *Aerosp. Sci. Technol.*, vol. 7, pp. 611–619, Dec. 2003.
[3] R. Lernbeiss and M. Plöchl, "Simulation model of an aircraft landing gear considering elastic properties of the shock absorber," *Proc. Inst. Mech. Eng. K, J. Multi-body Dyn.*, vol. 221, pp. 77–86, Mar. 2007.
[4] M. Karpel and E. Presente, "Structural dynamic loads in response to impulsive excitation," *J. Aircr.*, vol. 32, pp. 853–861, Jul. 1995.
[5] W. R. Krüger and M. Morandini, "Recent developments at the numerical simulation of landing gear dynamics," *CEAS Aeronautical J.*, vol. 1, pp. 55–68, Sep. 2011.
[6] M. Price, S. Raghunathan, and R. Curran, "An integrated systems engineering approach to aircraft design," *Prog. Aerosp. Sci.*, vol. 42, pp. 331–376, Jun. 2006.
[7] J. A. Jiménez, J. A. M. Merodio, and L. F. Sanz, "Checklists for compliance to DO-178C and DO-278A standards," *Comput. Standards Interfaces*, vol. 52, pp. 41–50, May 2017.
[8] D. D. Walden, *INCOSE Systems Engineering*. Hoboken, NJ, USA: Wiley, 2015.
[9] *Guidelines for Development of Civil Aircraft and Systems*, SAE, Warrendale, PA, USA, 2010.
[10] R. H. Daugherty, *A Study of the Mechanical Properties of Modern Radial Aircraft Tires*. Washington, DC, USA: NASA 2003.
[11] F. E. Cook and B. Milwitzky, "Analysis of landing gear behavior," NACA, Washington, DC, USA, Tech. Rep. 1154, 1953.
[12] M. Karpel, A. Shousterman, C. Maderuelo, and H. Climent, "Dynamic aeroservoelastic response with nonlinear structural elements," *AIAA J.*, vol. 53, pp. 3233–3239, Sep. 2015.

LINWEN LI received the Ph.D. degree in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2013. He has two years of visiting research experience for his joint Ph.D. study with the Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA, from 2010 to 2012. In 2013, he joined the Shanghai Aircraft Design and Research Institute, COMAC, in which he currently holds a senior engineer position. From 2017 to 2018, he was a Visiting Research Fellow with the Faculty of Industrial Engineering and Management and the Faculty of Aerospace Engineering, Technion–Israel Institute of Technology, Haifa, Israel. His research interests include advanced manufacturing processes on metal removal machining, model-based systems engineering (MBSE), and model-based design (MBD) interaction for aircraft design. He was a recipient of the Hiwin Doctoral Dissertation Award, in 2014, and the Chinese Government Fellowship Award for the High-end Equipment Talent Program, in 2017.

NATALI LEVI SOSKIN received the B.Sc. and M.Sc. degrees in information system engineering from the Technion–Israel Institute of Technology, in 2007 and 2009, respectively, where she is currently pursuing the Ph.D. degree in information management engineering under the supervision of Prof. D. Dori. Her M.Sc. research was in the area of information retrieval and concerned modeling uncertainty in *ad hoc* retrieval using multiple relevance models, published in part at ICTIR09. She has a decade of industrial experience as a Software Developer and an Information Systems Engineer at Intel and the Israeli Air Force. Her research concerns narrowing the gap between system and software engineering by integrating conceptual and computational modeling.

AHMAD JBARA received the Ph.D. degree in computer science from Hebrew University, Jerusalem, Israel, in 2016. For two years, he was a Researcher and an Adjunct Lecturer with the Enterprise Systems Modeling Laboratory, Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology. He was also a Faculty Member with the Computer Science School, Netanya Academic College. Since 2018, he has been with the Computer Science Department, University of Connecticut, USA. During the course of his Ph.D. studies, he coined the term ‘‘code regularity’’ and modeled its effect on comprehension. Insights from his M.Sc. thesis were adopted in an early version of Internet Explorer. His research interests are in the fields of program comprehension, code complexity metrics, code visualization, and conceptual modeling using object-process methodology (OPM).

MOTI KARPEL received the B.Sc. degree from Technion, in 1971, the M.Sc. degree from Tel Aviv University, in 1975, and the Ph.D. degree from Stanford University, in 1980. From 1971 to 1977 and from 1980 to 1988, he was with Israel Aircraft Industries, where he was the Head of the Dynamics and Loads Department. He was a Senior NRC Researcher with NASA Langley, from 1987 to 1988, and then joined Technion, where he held the Sanford Kaplan Chair of aerospace engineering, from 1999 to 2016, in the area of structural dynamics and aeroelasticity. He held visiting professor appointments at UCLA, GaTech, ITA, Imperial College, RMIT, and TU Delft Universities. He is currently a Professor Emeritus with Technion–Israel Institute of Technology. He is the Founder and the Technical Manager of Karpel Dynamic Consulting (KDC), Israel. He has developed several commercial codes for linear and nonlinear aero-servo-elastic dynamic loads and stability analysis, which are used at major aerospace industries, such as Airbus and COMAC. He has supervised 23 graduate students and has published 61 papers in refereed journals and more than 120 conference papers. He is a Fellow of AIAA.

DOV DORI (F'17) has intermittently been a Visiting Professor with the Massachusetts Institute of Technology, where he holds an adjunct position with the Systems Design and Management Graduate Program. Since 1993, when he invented object-process methodology (OPM), he has been central to the field of model-based systems engineering. OPM is the first conceptual modeling language and methodology recognized as ISO 19450:2015. He is currently a Professor of software and systems engineering and the Head of the Enterprise Systems Modeling Laboratory, Technion–Israel Institute of Technology. He has authored over 380 highly cited publications. He has supervised over 50 graduate students in Israel and USA, of whom 13 are faculty members in several countries. His 2002 book on OPM has been cited over 600 times. His book on model-based systems engineering is the basis for edX MBSE101 MOOC. His research has been funded by public and industrial sponsors, including the EU, Whirlpool, and Airbus. He has chaired nine international conferences. He was an Associate Editor of the IEEE TPAMI and *Systems Engineering* and a Co-Chair of the IEEE Society on SMC, TC on Model-based Systems Engineering. He has received various research and innovation awards and is a member of the Omega Alpha Association–International Honor Society for Systems Engineering. He is a Fellow of INCOSE and IAPR.

• • •