Research in Engineering Design (2019) 30:271–289
https://doi.org/10.1007/s00163-019-00308-6

ORIGINAL PAPER

Towards an integrated process model for new product development with data-driven features (NPD^3)

Yunpeng Li[1] · Utpal Roy[1] · Jeffrey S. Saltz[2]

Received: 31 August 2017 / Revised: 12 January 2019 / Accepted: 19 January 2019 / Published online: 29 January 2019

Abstract

New information and communication technologies are changing the way products are developed, manufactured, serviced and managed over the product's lifecycle. Today's smart products not only consist of their physical components, but are also endowed with intelligence. Data and the capabilities to process data into knowledge and eventually decisions have become critical components of the product itself and of the process to develop/operate the product. This paper investigates how engineers and a new functional role, data scientists, can effectively collaborate in a mixed team for new product development with data-driven features (NPD^3). We focus on the concept development stage, typically the fuzziest phase of product development. In this paper, an integrated process model is explored by revisiting the traditional new product development (NPD) process model as well as the knowledge discovery and data mining (KDDM) process model. Then a case study of the development of an application-specific unmanned aircraft system (UAS) is used to examine the proposed model.

Keywords Smart product development · Concept development · NPD · KDDM · Data-driven product design · UAS development

1 Introduction

The transformative power arising out of the fusion of information and communication technologies including sensor networks, big data analytics and cloud computing has changed the way a product is developed, manufactured, serviced and managed throughout the product's lifecycle. Products are getting smarter with the capabilities to perform reasoning based on known knowledge and to learn new knowledge from past experience (Li et al. 2015). With sensors and complicated algorithms, a household thermostat can, for example, autonomously establish a mathematical model that captures a building's inside thermal dynamics, without prior knowledge about the building characteristics such as its size, layout, leakiness, and HVAC system (Nest Labs 2012). Equally interesting, an unmanned aircraft vehicle can establish an occupancy map of its environment and can sense and avoid obstacles. As these two examples demonstrate, data and the capabilities to process data into knowledge and decisions have become critical components of the product itself and of the process to develop/operate the product.

The need for smart products to monitor, control and provide adaptation capabilities sets them apart from traditional products. The coordination needed across product design, cloud operation, service improvement, and customer engagement is continuous and never ends, even after the sale (Porter and Heppelmann 2015). Often times, the use of sensors within smart products provides the data needed for intelligence. Data analytics provides the tools and technologies needed to increase the intelligence of the device (Li et al. 2015). New data-centered product design and development paradigms have been emerging to inform the traditional processes and for the development of data-driven products. In the data-informed design paradigm, data can be utilized to reveal patterns and trends to drive innovation, measure product performance, and incrementally improve the product experience (Pavliscak 2015). As a result, a data-centered design approach can improve the development and operations of the product. On the other hand, data can be the "material" being processed by machine-learning algorithms

✉ Yunpeng Li
yli179@syr.edu

[1] Department of Mechanical and Aerospace Engineering, Syracuse University, Syracuse, NY 13244, USA

[2] School of Information Studies, Syracuse University, Syracuse, NY 13244, USA

to produce data-driven products (e.g. predictive or prescriptive analytics models), which in turn, can generate more data (Patil 2012; Dhar 2013). In this case, data are used to create data-driven machine-learning features within a smart product. The two paradigms have been collectively used in the development of modern smart mechatronics systems. For instance, automotive companies are employing these data-centered design techniques for the development of a car's autopilot capability as well as to improve the car's reliability (Geiger and Sarakakis 2016).

From both the product and process perspectives, data-driven modeling has emerged as a complement or replacement to the traditional knowledge-driven approaches. Data-driven modeling focuses on using advanced machine-learning methods in building models that capture physical behaviors (Solomatine and Ostfeld 2008). It is beneficial in situations where there is a considerable amount of data available and where it is difficult to build adequate knowledge-driven models, due to the lack of understanding of the underlying physical phenomenon and/or the difficulty in constructing a mathematical model of such a phenomenon. Consequently, this new discipline, Data Science, and the new experts, Data Scientists, need to be incorporated into the product development team (Porter and Heppelmann 2015).

The rest of this paper is as follows. First, Sect. 2 provides more details on our motivation and our research goal to explore the new paradigm of product development with data-driven features. Then Sect. 3 provides the background contexts of previous work in modeling the physical product development and data analytics processes. Section 4 explains our integrated process model, NPD^3, to incorporate the traditional new product development process model. Section 5 reports a pilot study that was conducted on a smart unmanned aircraft system development project that utilized the NPD^3 approach. Section 6 discusses the observations from the case study and reflects the new model's implications. And finally, Sect. 7 concludes our findings as well as reviews some research limitations and potential next steps.

2 Motivation

To better understand the motivation of our research, we first review the Nest Thermostat's auto-schedule feature and suggest the need for an integrated process model. With respect to the Nest's energy-efficient thermostat, the primary customer need is to achieve greater energy savings while maintaining the user's comfort. However, literature has reported that many residential thermostats fail to achieve energy savings even though they can be automated via programming because users tend not to use the feature (Peffer et al. 2011). The Nest Thermostat was the first self-learning thermostat that implemented a smart feature called Auto-Schedule (Lohr 2011) to fill the gap. It employs a sophisticated machine-learning algorithm that can automatically learn a user's preferred temperature profile as well as his/her schedule. This auto-schedule feature together with its supporting smart features (Auto-Away detection, Time-to-Temperature estimation, etc.), as well as the underlying data and computing infrastructure, form a smart ecosystem named Nest Sense.

A set of technical reports explain the details of the auto-schedule feature development and improvement (Nest Labs 2012, 2013, 2014). Figure 1 shows the Nest Thermostat's product structure including its physical components, embedded analytics features, and cloud-based analytics services. The auto-schedule's decision-making process and its data dependency are presented on the right side of Fig. 1.

According to Nest Labs (2012), the first-generation auto-schedule feature was developed via simulation. The simulation model consisted of physics-based models (including heat transfer model, air infiltration model and weather model, heating/cooling equipment model) and data-driven analytics models (auto-away, auto-schedule, time-to-temperature) to capture the dynamics of the environment in which the thermostat had been installed. Three years later after the first release, an Enhanced Auto-Schedule feature was released. This upgrade was a result of utilizing the accumulated actual usage data collected from many houses across different climate regions, thus more accurately captured thermal dynamics and users' behaviors (Nest Labs 2014). This enhanced auto-schedule feature had been upgraded on all three generations of Nest Thermostats in service ever since 2011, without introducing new hardware components.

The Nest Thermostat example shows data had become a critical factor that drives the development, operations, and improvement activities related to the product itself and its ecosystem. The Nest development team needed to co-develop the physical architecture (e.g. HVAC control) and the data architecture (e.g. a home model) to achieve optimal solutions for energy savings. Each embedded data analytics feature (e.g. auto-schedule) can be seen as a product part/component rather than an operational function. This is because the analytics feature adapts for each individual-installed instance rather than an aggregate function for a fleet of products. Consequently, data analytics is no longer an operational process, but rather, a product development activity that introduces new product features.

That is to say, product engineers and data scientists (if this role can be decoupled from or newly introduced into the development team) need to work together to formulate the problem, explore, screen, and evaluate the potential concepts, and eventually select one or more optimal concepts to finalize the product specification (Fig. 2). The conceptual question is how to decompose the tasks for data scientists

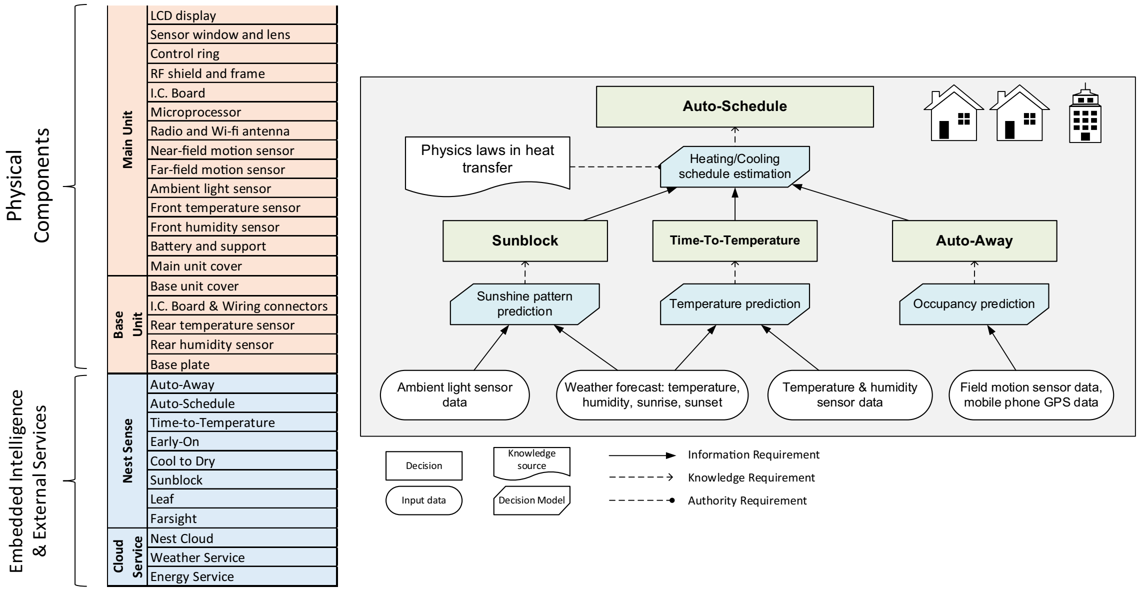
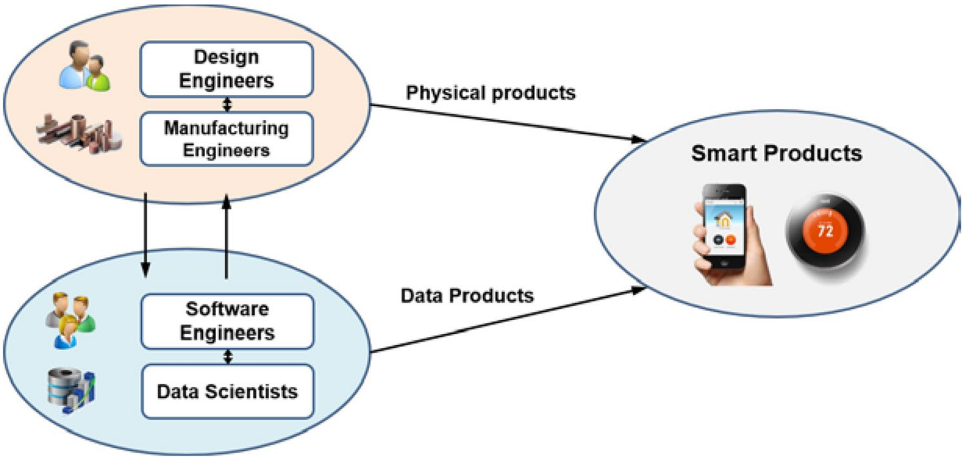


Fig. 1 The Nest Thermostat's Auto-Schedule self-learning feature

Fig. 2 A multidisciplinary team to develop smart products



from a product development process? In another words, what are the key tasks that the engineers in a physical product development team and the data scientists in a data product development team need to conduct? Accordingly, the following research questions should be addressed: (1) which tasks need to be coordinated across the two team groups; (2) when and what information needs to be exchanged between the two groups to collectively achieve the product development; and (3) what are the patterns and characteristics of their interactions.

To answer these questions, we revisit the existing process models for physical product development, software development, and data analytics since each one has prescribed the common activities used in many practical projects. The standard steps and activities prescribed in these existing models provide an initial view of how engineers or data scientists individually work. We then hypothesize the potential collaboration points by aligning and comparing these models. This analysis helps to derive an initial integrated model that for both engineers and data scientists. We then apply the hypothesized model to a real-world smart product development case. We develop an information decomposition framework to qualitatively categorize the observations of the interaction patterns within the case study, which leads us to achieve a theoretical framework for presenting the detailed interaction contents of the information flows. The interaction patterns and information contents complement our initial view of the integrated model that depicts the high-level key tasks and information flows.

To start, we focus on investigating the Concept Development stage of a new product development process. We believe an effective collaboration between engineers and data scientists in the front-end of the process will help avoid wasteful rework in the downstream processes and will enable the creation of better products that maximizes the potential of both the physical and analytical components of the product.

3 Product development process models

Numerous process models have been proposed and adopted to understand, improve, and support the design and development processes for physical products as well as for software products. These process models either define the project structures at the macro-level, end-to-end flows of tasks at the meso-level, or individual process steps and their immediate contexts at the micro-level (Wynn and Clarkson 2018). There is no means one individual model could cover all the necessary tasks and activities for a product development project. The practitioners have to select and adapt appropriate models for their needs. Since our target user roles are engineers and data scientists, we explore the New Product Development (NPD) process models for our baseline engineering process and the knowledge discovery and data mining (KDDM) models for the data science process. Below, we describe these models and then discuss how we think about incorporating KDDM into NPD.

1. Physical product development process: New Product Development (NPD)

New product development (NPD) transforms a market opportunity into a product (tangible or intangible) that is available to the market. There are two types of process models that are typically used in the traditional product development process (PDP). A sequential process model, also known as linear or waterfall, is a stage-gate-based process that has dominated in the manufacturing industry for several decades and is also often used for development of large-scale software systems.

Many NPD models have been adapted from the Cooper's Stage-Gate model (Cooper 1994, 2008) that typically consists of a series of stages followed by gates (the middle lane in Fig. 3). The prescribed stages and the criteria for transiting from one stage to the next provide useful guidelines to practitioners using the process. There are many versions of this process, such as the Ulrich and Eppinger's model, which is one of the well-adopted stage-gate models for physical product development. It consists of six high-level stages: Planning, Concept Development, Sub-System Design, Detail Design, Testing and Refinement, and Production Ramp-up (Ulrich and Eppinger 2012), see the upper lane in Fig. 3.

An alternative process model is a spiral process that incorporates cross-phase iterations (Unger and Eppinger 2009). The spiral process is commonly used in the software industry in the form of an agile methodology. For example, an agile scrum methodology typically consists of a number of short development cycles (2–4 weeks) undertaken by a dedicated project team.

The trend of mixing agile and stage-gate processes has been seen recently in manufacturing companies (Karlström and Runeson 2005, 2006; Cooper 2014, 2016), particularly in high-tech companies developing large-scale mechatronics that consist of mechanical parts, electronic parts, and software (Eklund and Bosch 2012; Eklund et al. 2014; Conforto and Amaral 2016). The agile–stage-gate hybrid model combines the predictability and planning that is typically desired within manufacturing physical products with the dynamic capabilities of modern agile software development. This

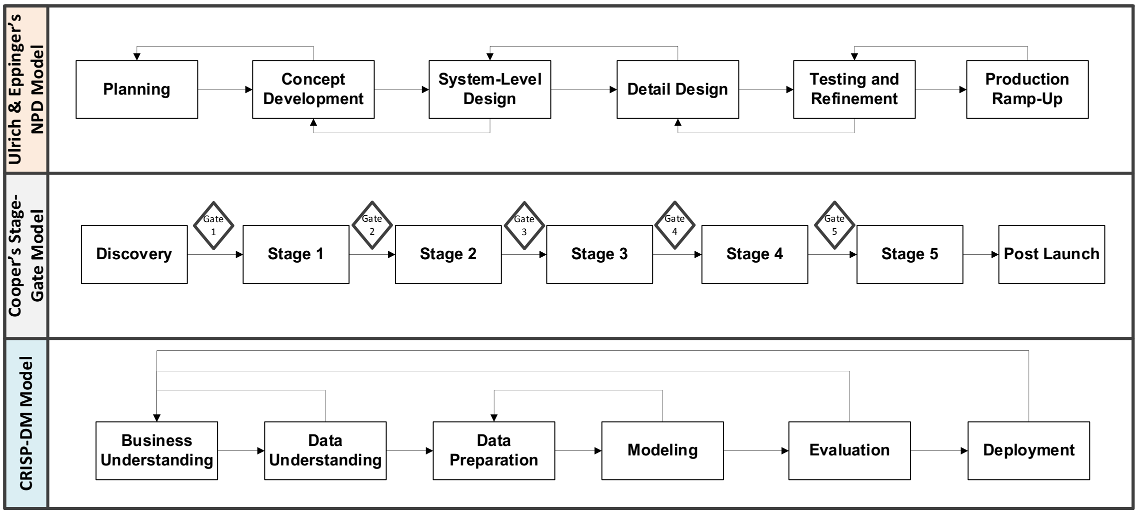


Fig. 3 The Cooper's generic Stage-Gate model, the Ulrich and Eppinger's NPD model, and the CRISP-DM model

agile–stage-gate model results in faster product releases, as well as better handling of changing customer needs, and improved team communication and morale (Cooper 2016). However, there are some challenges causing manufacturers to be hindered in the adoption of agile practices. The primary difficulty is that the development of a physical product cannot be easily incrementalized, in that creating a potentially releasable, working product in a short-sprint is not usually feasible. Furthermore, developing a mechanical part often includes developing and investing in very expensive manufacturing tools with long lead times, which can expand the development cycle for the product to 12 months or longer.

For these reasons, agile methodologies are currently employed mainly in the development and testing phases of a product development project. The overall product development approach at an organization level is still governed by a stage-gate model (for project management) or single-cycle V-model (for systems engineering). Hence, since our focus is on the concept development phase of the project, we leveraged a stage-gate model.

2. Data product development process: knowledge discovery and data mining (KDDM)

There has been a trend in the data science community to formalize data analytics projects as a Data Product development process. A data product is defined as a concrete component that facilitates the end goal analysis through the processing of data (Patil 2012). This product perspective suggests that data analytics is indeed a production process for producing data products, taking data as materials, turning data into usable knowledge models, and delivering results based on data. Data product development (DPD) produces software-like but data-centered products (e.g. data processing pipelines, statistics and machine-learning algorithms, and mathematical analytics models). However, the data analytics process is different from the traditional software development process because of the requirement to monitor and tune the model in short iterations and the fact that it is difficult for data scientists to know a priori what will be found when "exploring the data" (Saltz 2015). However, similar to software development, the data analytics process is iterative by its nature, and data scientists require constant revalidation of the problem, data sources, and outcomes.

Formal process models for data analytics projects originated from the knowledge discovery and data mining (KDDM) community. CRISP-DM (CRoss-Industry Standard Process for Data Mining) is one of the more successful process models that has been adopted by both industry and academia (Kurgan and Musilek 2006). CRISP-DM is a waterfall model that prescribes six high-level phases (the lower lane in Fig. 3) to formally describe a data analytics project and each phase is further decomposed into several key tasks and deliverables (Shearer 2000). The CRISP-DM's six high-level phases—Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment—appropriately capture the necessary lifecycle stages for data science activities (Li et al. 2015). This provides the possibilities to align the KDDM tasks with the NPD tasks to formulate a data-driven product development process, with appropriate adaptation and complementation of both models.

3. Incorporating KDDM in NPD

Most engineering design models prefer sequential models, focusing on an individual domain rather than considering interactions with other domains from a system perspective (Gericke and Blessing 2011). Integrated product development (IPD), or concurrent engineering (CE), is an effective means to address overlaps and interactions between multidisciplinary activities in the new product development process, increasing the need to coordinate and be compensated through other aspects of the NPD process (e.g., integrated tools), product definitions (e.g., incremental development), organizational context (e.g., reduced task specialization), and teaming (e.g., cross-functional teams) (Gerwin and Barrowman 2002). These traditional models have not explicitly addressed how to do data analytics during those processes. Systems engineering, another effective approach for developing multidisciplinary, large-scale, complex systems recently introduced the data-centric perspective (Wheatcraft et al. 2017). The concept is focused on the formalized use of a common, integrated dataset to support concept maturation, requirements analysis, design, analysis, verification and validation activities. This integrated dataset represents the work product and the underlying data and information generated during each lifecycle phase of the product. Similar to IPD/CE, the systems engineering framework has not explicitly prescribed a data analytics process.

From a data-centered perspective, an integrated product development process can be seen as an information-processing system or a decision production system, in which a network of stakeholders carries out various activities to process the development information, formulating specifications, concepts, and design details (Ulrich and Eppinger 2012). The process concludes when all the information required has been created and communicated, as well as when the key decisions have been made within the project time and budget constraints (Herrmann and Schmidt 2002; Krishnan and Ulrich 2011). This perspective implies there have been data analytics tasks embedded within the NPD process. Indeed, incorporating a KDDM process into an NPD process presents two further challenges.

The first challenge is related to the current data analytics practices as conducted in manufacturing firms. The data vary significantly across a product's full lifecycle (Kassner et al. 2015). The product, production, and service-related data are available in various manufacturing information systems (e.g., PLM, MES, and ERP systems) (Roy et al. 2014), but might also reside in an external supply chain partner's system. There can also be an inability of big data processing within manufacturing firms due to the limitations of IT resources within those firms (Sun et al. 2017). For example, few manufacturing experts are familiar with modern big data analytics techniques. If the data analytics tasks that have previously been embedded in the engineering processes could be decoupled, it would be more efficient for those tasks to be done by a dedicated data science team. The question is, what are those embedded tasks?

The second challenge is related to the natural latency between the physical product development activities and the data product development activities. This is because the development of accurate analytics models greatly relies on new data generated as part of the physical product development process and there is an inevitable time lag between these development processes (Li et al. 2015). In other words, while the NPD and KDDM processes both follow similar high-level stage sequences, there is no systematic way to synchronize the two sides' activities. Consequently, the different cycle times of physical product development and data product development can lead to less optimal solutions where issues are solved in software or data analytics (upgrade to the product) even though they would have been better solved in physical design (new generation of product), or vice versa. Understanding the interaction patterns of engineers and data scientists would be beneficial to the integrated decision-making process design, which in return facilitates a better system architecture design, for the development of a smart product.

4 An initial view of the NPD^3 model

Taking the abovementioned information-processing perspective, the product development system becomes an information network. This product development information network usually consists of three levels of information-processing units (Collins et al. 2008; Distanont et al. 2012): (1) the overall structure—the product development process as a whole is a single entity of tasks that share information; (2) the subgroup—the groups of tasks that interact more with each other than with other tasks in the product development process; and (3) the individual tasks—the key tasks that are identified based on their relational roles as information transmitters (coordinator, gatekeeper, representative, liaison, or consultant).

To maximally leverage the existing models, we need to (1) identify the three levels of units already prescribed in the standard NPD model and the CRISP-DM model; (2) identify the prescribed tasks for individual roles. More specifically, in the NPD model we focus on the tasks prescribed for project manager, design engineer, and manufacturing engineer; in the CRISP-DM model we focus on the tasks prescribed for data scientists. Note that this paper employs the business process model and notation (BPMN) and decision model and notation (DMN) conventions to represent process workflows and the decision-making logics in a data-driven product. Compared to other process diagramming approaches such as business process execution language (BPEL) and Petri Nets, BPMN focuses more on participants, and controls their interactions and flow with events and decisions (Debevoise and Taylor 2014). BPMN and its companion, DMN, for modeling modular decision models, can be automated in a business process management system.

As mentioned previously, we focus on analyzing the concept development stage. The main engineering tasks prescribed in the NPD concept development stage include Investigate feasibility of product concepts, Develop industrial design concepts, Build/test experimental prototypes, Estimate manufacturing cost, and Assess production feasibility (Ulrich and Eppinger 2012). Design engineers usually fulfill the first three tasks and manufacturing engineers typically fulfill the last two tasks. According to Marbán et al. (2009), the tasks defined in the CRISP-DM that are relevant to concept development (for data products) are mainly in the Business Understanding and Data Understanding stages. We argue that the concept development should focus on the role of translating business needs into technical implementation specifications. Therefore, we align the CRISP-DM's Business Understanding stage with the NPD's Planning stage, and we only count the tasks defined in the Data Understanding stage as concept development activities for data products. These tasks include Collect data, Describe data, Explore data, and Verify data. It is noted there are implicit activities when exploring data: hypotheses modeling and testing (descriptive analytics), followed by discovering data mining opportunities (predictive analytics). These exploratory activities are analogous with the concept investigation and design activities in NPD, and shall be differentiated from the later Modeling stage of the CRISP-DM. Therefore, we explicitly add these activities in the Data Understanding stage and term them as Generate and test initial hypothesis, Investigate feasibility of predictive analytics, and Discover repeatable analytics services.

In summary, the engineering activities are grouped as Identify and design concepts, Build and test concepts, and Evaluate concepts for selection; similarly, the data science activities are grouped as Identify and collect data, Descriptive analytics, Verify data quality, and Investigate analytics concepts. This setting structures the time lag for data scientists' activities compared to the engineers' activities.

Specifically, in Fig. 4, the upper lane represents the engineering team, who focus on the engineering tasks for the development of the physical product. They translate the customer needs into the technical specification and use the technical specification to come up with the optimal solution for the physical design. Engineers usually employ well-established Design for 'X' principles (e.g. Design for Manufacturing and Assembly, Design for Environment, etc.) to evaluate and refine the product concept (Li and Roy 2018). The final specification includes a bill-of-materials of the physical components and target values of their properties. The Identify and design concepts task has larger information integration workloads while the Evaluate concepts for selection has larger information dissemination workloads.

The lower lane represents the data science team, who focus on data processing and analytical modeling tasks for the development of the data product. They translate the customer needs into the data specification and use the data specification to come up with the optimal data analytics solution. Since data quality greatly impacts the analytics results, there must be a gateway to go/kill the decision to the Investigate analytics concepts task. The final specification includes both the data specification and the analytics feature specification. Similarly, a bill-of-services shall be included if the analytics feature can be further decomposed into reusable services. Intuitively, the Identify and collect data is dominated by information collection workloads and the Investigate analytics concepts task is dominated by information dissemination workloads.

The middle lane in Fig. 4 represents the project management (PM) team, who follow a stage-gate-based NPD process. It takes a Mission Statement as input and produces the approved Development Plan. The first two tasks (Identify customer needs and Establish target specification) involve the marketing team, management team, customers, and other stakeholders. The engineers and data scientists participate in these preparation stages, and their collaboration is mainly brokered by the PM team. The dominance of outgoing flows indicates the information brokerage and dissemination role of these tasks. The design-build-test task group comprise of the core activities with which the engineers and data scientists work to collectively solve the problem. Detailed tasks are conducted in the individual team activities. Engineers and data scientists can use face-to-face communications if the organizational structure and geolocation allow, and the PM team can focus on ensuring the coordination of the tasks. This task group is also where the high rate of iteration takes place. The last task, Set final specification, again involves stakeholders from many disciplines to complete the development plan. The dominance of incoming flows indicates the information integration workloads of the task.

Note that the engineering and data science tasks are coordinated by the PM design-build-test tasks; hence, there are two implicit gateways (for project decomposition and integration) located before and after the design-build-test task group. It is also noted that test data from a simulation model, a physical prototype, or a field test can only be obtained after such a model/prototype has been built. Therefore, there is a message flow from the Build and test concept task to the Identify and collect data task of a later iteration for data scientists. In addition, the sequence flows across the discipline boundaries also carry the necessary message information; we do not draw explicit message flow symbols for a clearer representation.

5 Case study using the NPD^3 model

The initial NPD^3 model sets up the key tasks and main data/information flows, but we need to understand, in more detail, the content of these information flows and the team interaction patterns. In this section, we report on a project that utilized the NPD^3 approach to develop an unmanned aircraft system (UAS) that integrated advanced analytics within an unmanned aircraft vehicle (UAV) and its supporting systems, which we term a "Smart UAS".

Information for this case study consisted of weekly semi-structured observation notes. In addition, a product lifecycle management (PLM) system was deployed for the team to centrally store the project artifacts (e.g., project weekly meeting minutes, 3D models, simulation data) and this project documentation was also leveraged to analyze the case study.

We start with a brief overview of the generic UAS architecture and its data-driven needs, followed by the project requirements, team formation, and the data infrastructure to support the team collaboration. We then report on the concept design process, including discussion of the challenges faced by the project team, the concept testing that was performed, and how our NPD^3 approach was leveraged within the case study. In Sect. 6, we discuss in detail a theoretical framework for the information decomposition based on our observations within the case study.

1. The data-driven features for UAS

A UAS consists of five distinct elements (NATO 2012): (1) the Unmanned Air Vehicle (UAV) element includes the air frame, power system, and the avionics required for flight control; (2) the Payload element includes the sensor systems, associated recording devices, and associated control/feedback mechanisms; (3) the UAV Control System (UCS) element incorporates ground and air control systems for generating, loading, and executing the mission and to disseminate information to various command, control, communication, and intelligence (C4I) systems; (4) the Launch and

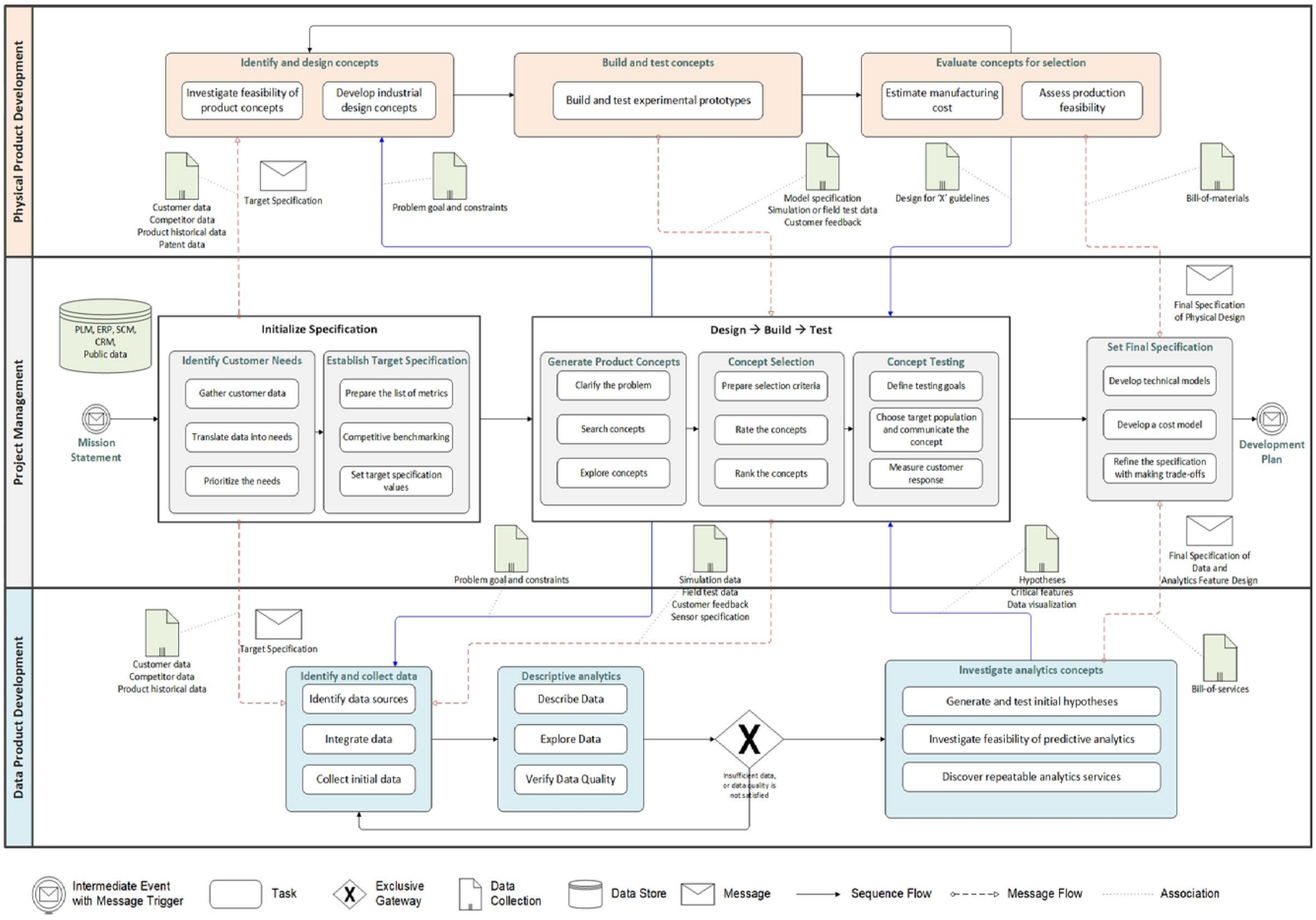

Physical Product Development
Project Management
Data Product Development
Identify and design concepts
Investigate feasibility of product concepts
Develop industrial design concepts
Build and test concepts
Build and test experimental prototypes
Evaluate concepts for selection
Estimate manufacturing cost
Assess production feasibility
Customer data
Competitor data
Product historical data
Patent data
Target Specification
Problem goal and constraints
Model specification
Simulation or field test data
Customer feedback
Design for 'X' guidelines
Bill-of-materials
PLM, ERP, SCM, CRM, Public data
Mission Statement
Initialize Specification
Identify Customer Needs
Gather customer data
Translate data into needs
Prioritize the needs
Establish Target Specification
Prepare the list of metrics
Competitive benchmarking
Set target specification values
Design → Build → Test
Generate Product Concepts
Clarify the problem
Search concepts
Explore concepts
Concept Selection
Prepare selection criteria
Rate the concepts
Rank the concepts
Concept Testing
Define testing goals
Choose target population and communicate the concept
Measure customer response
Final Specification of Physical Design
Set Final Specification
Develop technical models
Develop a cost model
Refine the specification with making trade-offs
Development Plan
Final Specification of Data and Analytics Feature Design
Problem goal and constraints
Simulation data
Field test data
Customer feedback
Sensor specification
Hypotheses
Critical features
Data visualization
Customer data
Competitor data
Product historical data
Target Specification
Identify and collect data
Identify data sources
Integrate data
Collect initial data
Descriptive analytics
Describe Data
Explore Data
Verify Data Quality
X
Insufficient data, or data quality is not satisfied
Investigate analytics concepts
Generate and test initial hypotheses
Investigate feasibility of predictive analytics
Discover repeatable analytics services
Bill-of-services
Intermediate Event with Message Trigger
Task
Exclusive Gateway
Data Collection
Data Store
Message
Sequence Flow
Message Flow
Association

Fig. 4 An integrated process model for new product development with data-driven features (NPD3)—concept development

Recovery element incorporates the functionality required to safely launch and land the UAV; and finally (5) the Data Link element, which enables ground–air communication or air–air communication.

The data-driven nature of a smart UAS arises from its transition from an automated system to an autonomous system (Li et al. 2017). The autonomy of an UAS is defined as the UAS's own abilities of sensing, perceiving, analyzing, communicating, planning, decision-making, and acting/executing, to achieve its goals as assigned by its human operator(s) through a designed human–robot interface or by another system that the UAS communicates with (Huang 2008). The autonomy enabling functions for a UAS can be grouped into three subsystems: navigation, guidance, and control (Kendoul 2012). Navigation is the process of monitoring and controlling the movement of an air vehicle from one place to another. It is a highly data-intensive process involving data acquisition, analysis, extraction and inference of information about the vehicle's state and its surrounding environment with the objective of accomplishing the assigned mission successfully and safely. Guidance is the driver of the UAS that exercises the planning and decision-making functions to achieve the assigned mission or goal. It takes inputs from the navigation system and generates reference trajectories and commands for the flight control system. Finally, control is the process of manipulating the inputs to a dynamical system to obtain a desired effect on its outputs without a human in the control loop.

Figure 5 depicts the core elements of an autonomous UAS's generic architecture, which consists of its physical architecture, autonomy architecture, cyber-physical interfaces, and the supporting subsystems.

2. The UAS requirement, team formation, and project management

The smart UAS was designed for a Water-Quality-Sampling application that was requested by a civil engineering scientist. The usual practice in this area is to collect small water samples for lab analyses because many water properties cannot be measured in the field (Ore et al. 2015). If the properties can be measured in the field, they require an onsite monitoring system or a suitable vehicle to carry the instruments. In our case, the scientist requested a UAS to measure the water properties including temperature, pH, dissolved oxygen, etc. A UAS platform could access hazardous environments, be more flexible than an onsite water monitoring system, and be faster than other vehicles (e.g. a boat). Most importantly, if properly designed, a UAS platform could be a cost-effective solution with the capability to adapt itself to conduct different missions. The overall requirements and the initial system specification are shown in Table 1.

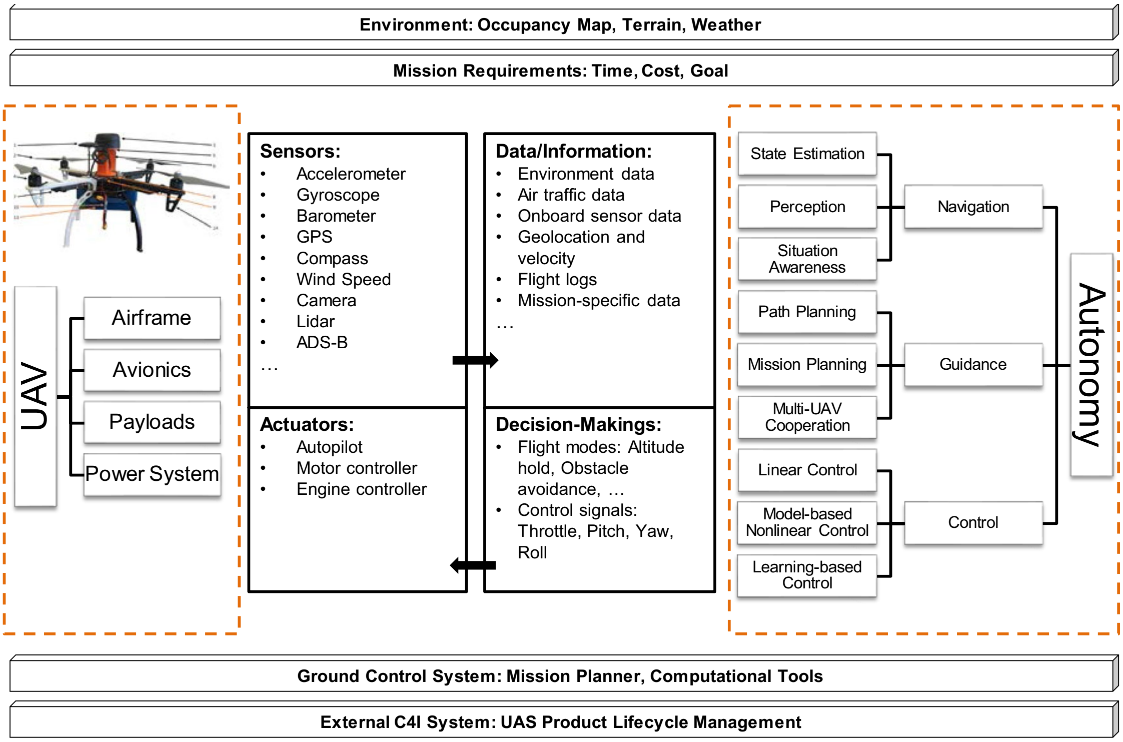


Fig. 5 A generic architecture of an autonomous unmanned aircraft system (UAS) (Li et al. 2017)

Table 1 The UAS requirement and the Target specification

Requirement	Target specification
Is lightweight to be carried and operated by a single scientist	The total weight is less than … kilograms with all the necessary payloads
Is cost-effective	The total cost is less than … dollars
Can hover over the water area	The UAV can hover 1 m above the water surface without moistening the onboard payloads
Is safe and has certain levels of autonomy	The UAV can detect-and-avoid a static or dynamic obstacle within a radius 5 m surrounding it
Can measure a range of water-quality properties at predefined locations	The UAV can measure pH, temperature, dissolved oxygen using onboard sensors The accuracy of measurement should fall between … and …
Can collaborate with other UAVs to achieve a mission	The UAV can work with at least one another UAV to simultaneously measure the water-quality properties at a predefined location without sacrificing the safety
Can quickly adapt to different water-sampling missions	The UAV can replace with different sensors to sample different water-quality properties

There were twelve people working on this smart UAS project. As shown in Table 2, the team was comprised of researchers on the mechanical, electrical and data groups, as well as a remote pilot, a project manager, and an industry expert. The authors of the paper were part of the team to help building the data infrastructure, providing data analytics guidance, and coordinating the project management. In the project kickoff meeting, the authors presented the NPD^3 diagram of Fig. 4 to the project team and explained the NPD model, the CRISP-DM model, and the integrated model. The NPD^3 model provided a common language and guidance to both engineers and data scientists, who otherwise are not familiar with the process used by their counterparts. The authors then documented the observations via weekly semi-structured notes throughout the remaining project time.

A data model was developed to capture the metadata of the generic elements and their relationships of the UAS architecture shown in Fig. 5. The data model derived was based on the concept of the smart component data model (Li et al. 2015). This abstract model facilitated data storage, access, exchange, and tracing of all the data generated throughout the project. The core classes of the UAS data model are described as follows:

- *PLM generic item* The root class of the PLM system; all other classes inherit from this class and the children of this class.
- *Physical component* The physical components of a product to form its body. The classes for the overall air vehicle, the airframe and propellers, the avionics, the

Table 2 The multidisciplinary team

Role group	Role description	Team members
End user	Providing the domain knowledge regarding the water-quality monitoring application	One professor from civil engineering
Mechanical design and manufacturing	Designing the UAS mechanical parts and their configurations 3D modeling and 3D printing of the custom-made parts	Two professors from mechanical engineering One PhD student and one Master student from mechanical engineering
Electrical design and software development	Designing the UAS autopilot, communication and control, and autonomy algorithms	One professor from computer science and one professor from control engineering
Data architecture and analytics	Designing and implementing the data management architecture and data processing platform Conducting data processing and data analytics	Two PhD students from mechanical engineering with the background of industrial engineering and data science
Pilot/Operator	Operating the UAV for test flights and mission flights	One FAA-certified UAS remote pilot
Project management	Overseeing and coordinating the project	One principal researcher of the project
Project sponsor and industrial advisor	Providing advice and feedbacks from the industrial perspective	One industrial expert from sensor industry

payloads, and the power systems are inherited from this class.

- *Analytical component* The analytical components of a product to implement its intelligence. All the autonomy-related functions (navigation, guidance, and control) can be implemented in different derivations of this class. Business rules, a specific case of analytics models, also inherits from this class and implements the regulatory rules.
- *Dataset* The datasets that have been extracted, aggregated, cleaned, and structured from various sources of raw data. It can be a training or test dataset and provides the context to the analytics models built.
- *UAS* An application-oriented UAS that is composed of certain physical components and analytical components and is compatible with a range of missions.
- *Mission plan* The operations of an individual UAV or a fleet of UAVs to fulfill the mission requirement.

3. The UAS development

There were not many UAS-based water-sampling applications available when this project was started. In the project preparation stage (the first 2 weeks), a large number of articles in the fields of infrastructure management, environment monitoring, and traditional water-sampling methods were reviewed and studied. The recent research topics on UAS were also explored from technical publications. For example, publications on the International Conference on Unmanned Aircraft Systems (ICUAS, http://www.uasconferences.com) during 2013–2016 timeframe indicate topics such as UAS applications, navigation, path planning, control architectures, and simulation were constantly the top research areas. Other data sources included the product specifications from UAV and sensor vendors, the patent database for water-sampling mechanism design, and government data regarding water-quality monitoring.

The sharing of this information, including the system requirements, literature analyses, and other publicly available information, together with the initial target product specification was coordinated by the project management team and able to be accessed by both the engineering and data analytics groups for concept development. In the early concept exploration stage, the project team met frequently to brainstorm the possible concepts, during which the domain knowledge had to be exchanged.

Each concept needed to consider a suitable configuration of the UAV hardware (air frame, avionics, payload, and power system), autonomy functions (state estimation, obstacle avoidance, etc.), and data communication methods. To leverage the potential of the data analytics, several questions were consistently asked by the team as they generated each new concept:

- Is the current knowledge sufficient to capture the real-world dynamics (for example, the water area the UAV will fly over)?
- If not, can the problem in hand be solved by a data-driven modeling approach, and with what hypotheses?
- What data should be collected and how often should the data be collected?
- What sensors should be used and what parameters are required?
- How to decompose the decision-making process of an autonomy function?
- What are the repeatable/reusable analytics services can be adapted for future applications?
- Where to implement the analytics services, onboard or offboard? What are the physical constraints?

These questions occurred across all levels of the concept development process. For example, it was difficult to preestablish a model for the target flight environment. The system needed to check the terrain, water surface, weather dynamics, and any possible surrounding obstacles. The establishment of such an environmental model needed significant effort to work with many external data service providers, for instance, the UTM (UAS Traffic Management) services. The algorithms to map the environment could be implemented either at the ground control station computer or onboard the UAV equipped with LiDAR sensors or vision cameras. Furthermore, these functions should work independently without affecting the water sampling, the main function of the UAS. This implies that the data infrastructure and communication protocols had to be co-developed with the UAS hardware and control software at the system architecture level. At the component level, a challenge the team faced was 'What if the UAS is used in a GPS-denied environment where the GPS signal is no longer available'. In this situation, two alternative concepts could be viable: (1) use other types of global positioning systems to provide the GPS-equivalent data; or (2) use a completely different localization method, for example, a vision-based or a LiDAR-based system, to predict the desired state variables. In the first case, another positioning system (such as GLONASS) providing the same GPS format data solves the problem, and the software and all the data-processing functions for state estimate are not necessarily changed. In the second case, additional sensors, processors, software, and data-processing functions have to be built into the design, which means the team would need to review and revise the system architecture.

Similar to the Nest Thermostat development, the team set up a hybrid simulation environment that incorporated a UAV simulator, communication hardware, and the PLM system as a data repository to automatically store the test mission plans and process the mission data so as to provide performance analysis and train the machine-learning models (Fig. 6). The simulator used the same autopilot controller firmware as was used by the real UAV so that the simulation settings could be used for field test flights. The data scientist could work with the simulation data to explore the data and build initial machine-learning models before any real data had be collected from the field flights. Then the validity of the models could be tested by the field tests. The data were used for either diagnosing the UAS performance (e.g. flying stability due to inappropriate tuning or signal interference) or for historical data to build predictive models (e.g. obstacle recognition and avoidance). The results of the data analytics were twofold: (1) feedback to the next design iteration to inspire a new design, and (2) analytics models directly improving the current concept. The latter is an interesting "self-improvement" effect, in that it is a unique characteristic in a data-driven product. For example, we can recall the smart thermostat case, where more installations and usage generate more data that could be used to improve the auto-schedule feature; similarly, more flight scenarios could enhance the UAV obstacle avoidance capability.

Several sample concepts are presented in Table 3. Both the engineers and data scientist had their domain-specific requirements. For example, a set of well-established Design for 'X' principles (e.g. design for manufacturing and assembly, design for sustainability) were used by the engineers to evaluate and refine the product concepts. The product bill-of-materials for physical components was critical to determine the selection of raw materials, manufacturing tools and processes, as well as the assembly/disassembly and recycling methods. Similarly, data scientists employed a set of measurements, including data quality, prediction accuracy, computational costs, the capability to incrementally update with new data, in order to screen the analytics models. A bill-of-data and bill-of-services for data analytics models were also critical to determine which data analytics

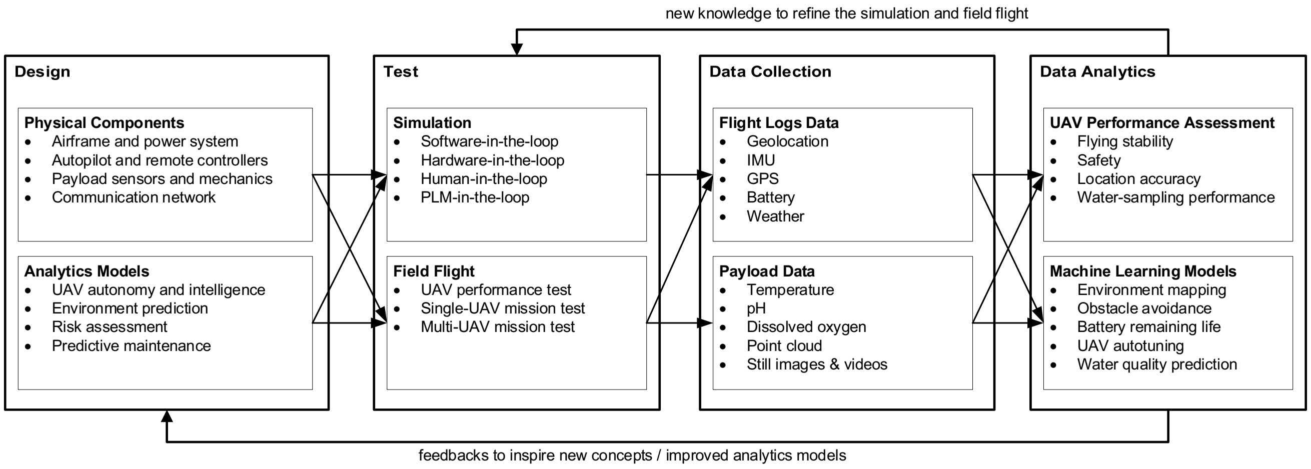


Fig. 6 The UAS design, simulation, test, and data analysis

Table 3 Several UAS concepts

Concept	Concept description	Characteristics
Concept #1	Quadcopter with autopilot ADS-B for sense-and-avoid Water-quality sensors Cloud data storage	Low cost Collaborative sense-and-avoid capability No to low-level autonomy
Concept #2	Quadcopter with autopilot and onboard computer LiDAR for detect-and-avoid Water-quality sensors Onboard and cloud data storage	Affordable Non-collaborative sense-and-avoid capability Low-level autonomy
Concept #3	Octocopter with autopilot and onboard computer ADS-B and LiDAR for detect-and-avoid High-precision water-quality sensors Onboard and cloud data storage	Expensive Collaborative and non-collaborative sense-and-avoid capabilities Medium-level autonomy High payload capacity

techniques should be employed in the downstream processes. Collectively, the criteria for concept ranking and selection took the functionality, level of autonomy, cost, degree of modularity, and regulation requirements into consideration. Here, the level of autonomy is an important criterion even though it may compromise the overall cost and the product modularity (because of redundant components and computation). The team chose the second concept that was overall affordable and was satisfied with the project requirement.

The final UAS specification included the hardware specification, software specification, data specification, and analytics model specification. System design and other later stages defined in the NPD and CRISP-DM were then followed.

6 Further discussion

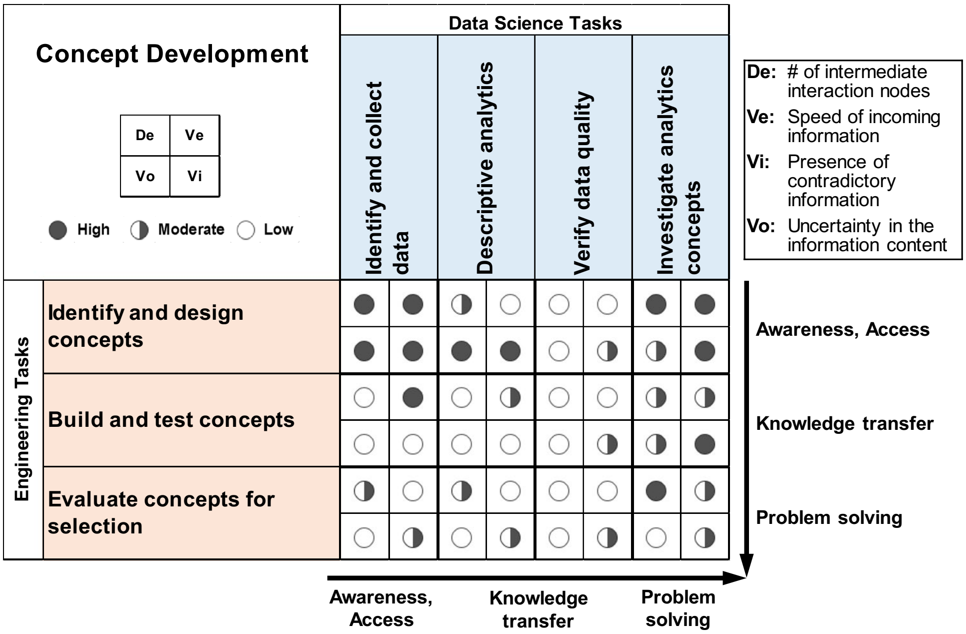
1. Observations of the team interaction patterns and characteristics

The UAS development project lasted for five months. The team interaction patterns are summarized and shown in Fig. 7. Specifically, the rows in the table represent the phases of the project from a data science perspective and columns represent the phases of the project from an engineering perspective, and each cell represents a possible set of interactions.

In analyzing the goals of the interactions, we followed Distanont et al. (2012), who noted that, in a collaborative product development network, one can view the interaction flows as one of four goals for the interaction—awareness, access, knowledge-transfer, and problem-solving. Figure 7 shows that as one moves further along the concept development process, the goal moves towards problem-solving. In the smart UAS project, a significant amount of interactions was needed to identify the data sources and interests during the early project phase. The multidisciplinary team was finally able to collectively deliver the water-sampling UAS platform with appropriate composition of physical components, compatible control software, and suitable data analytics pipeline.

Furthermore, in analyzing the characteristics of each information flow that occurred within our case study, we leveraged four attributes proposed by Krovi et al. (2003). The density (De) is defined by the number of intermediate interaction nodes. The velocity (Ve) refers to the speed of incoming information at an interaction node. The viscosity (Vi) reflects the degree of conflict due to presence of contradictory information components at the interaction node. The volatility (Vo) denotes the associated uncertainty in the information. At a high level, when the UAS project had to integrate and evaluate the various concepts, the presence of contradictory information increased because there had to be a compromise for the multiple performance measures. It was also observed that the speed of incoming information was initially high, then decreased but then increased later, once the simulation model started to generate data based on various trial settings. As the process progressed, more data and information were available and the design problem was more constrained; therefore, the problem became less uncertain.

With this framework, we can categorize the interaction for the flow of a specific phase combination. Below we describe the interaction patterns among the different combinations


Concept Development
De
Ve
Vo
Vi
● High
◑ Moderate
○ Low
Data Science Tasks
Identify and collect data
Descriptive analytics
Verify data quality
Investigate analytics concepts
Engineering Tasks
Identify and design concepts
Build and test concepts
Evaluate concepts for selection
De: # of intermediate interaction nodes
Ve: Speed of incoming information
Vi: Presence of contradictory information
Vo: Uncertainty in the information content
Awareness, Access
Knowledge transfer
Problem solving
Awareness, Access
Knowledge transfer
Problem solving

Engineering Tasks	Identify and collect data		Descriptive analytics		Verify data quality		Investigate analytics concepts	
Identify and design concepts	●	●	◑	○	○	○	●	●
	●	●	●	●	○	◑	◑	●
Build and test concepts	○	●	○	◑	○	○	◑	◑
	○	○	○	○	○	◑	◑	●
Evaluate concepts for selection	◑	○	◑	○	○	○	●	◑
	○	◑	○	◑	○	◑	○	◑

Fig. 7 Interaction patterns and characteristics between engineers and data scientists

of engineering and data science project phases. As a starting point, we define each attribute to have three levels: low, moderate, and high:

Identify and design concepts—Identify and collect data At the start of the project, a significant amount of interaction between engineers and data scientists was needed to identify the data sources and interests. The data sources included market surveys, technical publications, patent database, government data, and manufacturer/vendor whitepapers. The velocity of incoming data was fast, the data/information present had a high conflict, and uncertainty was also high. In short, all four parameters were at the high level.

Identify and design concepts—Descriptive analytics At this stage, the data had been collected, and the data science team was focusing on the data analysis. Hence, the interaction density was moderate and velocity was low. However, the viscosity and the volatility were high because the two groups had a different understanding with respect to the large amount of data from the different sources. For example, it was difficult for the data scientists to understand the meaning of each column presented in the flight log data.

Identify and design concepts—Investigate analytics concepts At this stage, the data science team started to generate analytics concepts which in turn affected the development of the physical concepts. The interaction density again became high, the velocity and viscosity were also high since more data and information had become available. For example, the localization function required data from different sensors for GPS-friendly environment versus the GPS-denied environment. The concept designs for the sensor systems and the analytics models were mutually affected. The volatility was kept low to moderate.

Build and test concepts—Identify and collect data, Descriptive analytics At these stages, the data sources were mainly from the simulation, field test, and the customer feedback. The data formats had been determined and the data stream processing could be automated to some extent. With both the physical and analytical concepts being built into the prototypes, descriptive analytics were conducted on various testing scenarios. Sensitivity analysis was conducted to identify the impacting variables on the product performance. The interaction density was, therefore, low; the velocity was high to moderate. There were low viscosity and volatility.

Build and test concepts—Investigate analytics concepts At this stage, the data science team refined the previous analytical concepts, to generate and test new analytical concepts for the next iteration. The interaction density and velocity were moderate to high, the viscosity was high but the volatility kept low to moderate.

Evaluate concepts for selection—Identify and collect data, Descriptive analytics At these stages, the product concepts had been filtered to a limited set, and the focus of data scientists had turned to a new iteration to collect data for product performance analysis—to provide guidance in building a closed-loop product operation for continuous improvement. The interaction density and viscosity were moderate; the velocity and volatility were low.

Evaluate concepts for selection—Investigate analytics concepts At this last stage, both groups determined the final concepts. The interaction density and viscosity increased again because of the integrated evaluation and there had to be a compromise across the multiple performance measures. For example, collecting more data was beneficial to the analytics model development; however, this implied the sensors and the controllers needed to work in higher frequencies. Thus, it had a negative impact on the battery power. The interaction velocity was moderate. The volatility was low because most uncertainties had been eliminated and there had been risk mitigation plan in place.

All engineering tasks—Verify data quality At these stages, the data had been cleaned, processed, and descriptive analysis results were generated and presented. Data scientists needed engineering experts or other stakeholders to verify the data quality to prepare the datasets for the following analytical concept development tasks. The interaction density, velocity, and volatility were low. Since this was always a go/no-go decision-making point, the viscosity was moderate to high.

2. A theoretical view on the decomposition of information content for individual/subgroup tasks

In this section, we discuss, from a theoretical perspective, the details of the information flows related to an individual task or group of tasks. To show these input and output flows, we employ an IDEF0-based notation which decomposes the information related to unit collaborative design activity into four categories: intra-disciplinary design information (I), cross-disciplinary design information (C), external design information (E), and design information output (O). This notation was originally proposed by Austin et al. (1999) termed as IDEF0v, to facilitate a collaborative building-design process, while the IDEF0 (Integrated computer aided manufacturing DEFinition for function modeling) technique was developed to better communicate and analyze manufacturing systems in an attempt to improve productivity.

The information content of each information flow is identified by revisiting the standard activities defined in the NPD and the CRISP-DM, as well as the activities we found during our smart UAS project. For instance, the intra-disciplinary input information for the Identify and collect data stage consists of historical data of the product and production, while the output information includes the concept classification tree and combination table. By this way, the information flows for the engineering and data science activities are elaborated in Figs. 8 and 9. The intra-disciplinary, cross-disciplinary, external, and output information for the engineers are encoded as $\mathbf{I_{eng}1{\sim}3}$, $\mathbf{C_{eng}1{\sim}3}$, $\mathbf{E_{eng}1{\sim}3}$, and $\mathbf{O_{eng}1{\sim}3}$. We explicitly encode the potential cross-disciplinary information received from the data scientists as $\mathbf{C_{eng}1{\sim}3\text{-}DST}$. On the data scientists' side, $\mathbf{I_{dst}1{\sim}4}$, $\mathbf{C_{dst}1{\sim}4}$, $\mathbf{E_{dst}1{\sim}4}$, $\mathbf{O_{dst}1{\sim}4}$, and $\mathbf{C_{dst}1{\sim}4\text{-}ENG}$ are the intra-disciplinary, cross-disciplinary, external, output information, and cross-disciplinary information explicitly from the engineers.

This information decomposition reveals several interesting factors of the information dependency between the engineering and data science groups. First, the external and cross-disciplinary information shows the shared information between the two groups (for example, product specification, customer feedback, and publicly available data), indicating that a common dedicated team, or a higher level project management team (if there is one), could help broker this information. In the smart UAS project, the project manager, the industrial advisor, and the end user indeed helped to coordinate tasks for the collection and dissemination of these shared data. The PLM implementation of the high-level NPD^3 model also helped the data/information sharing. Second, the cross-disciplinary information coming from the two groups indicates engineers and data scientists may need to directly communicate with each other for effectiveness and efficiency, suggesting that an appropriate organizational structure or geolocation arrangement between the two groups would be helpful. In our case study, the engineers, the data scientists, and the remote pilot were from different departments, which created some time schedule and communication challenges. For example, there were situations where the data scientists were waiting for new test data but the pilot was not available for field testing the new engineering design.

Furthermore, the information content in the cross-disciplinary flow not only needs to be aware of and accessed by each group, but also transfers domain-specific knowledge to the counterpart for collective problem-solving. The output information of each activity is not only for the next activity within the same discipline, but might also be consumed by the collaborative tasks in the other discipline since an effective process requires each subsequent task to maximize the utility of the stable information available from the previous task (Cooper 2014). However, it was observed that the analytical concept generation was always at least one step behind the physical concept generation unless the data could be obtained from an existing data source. This implies

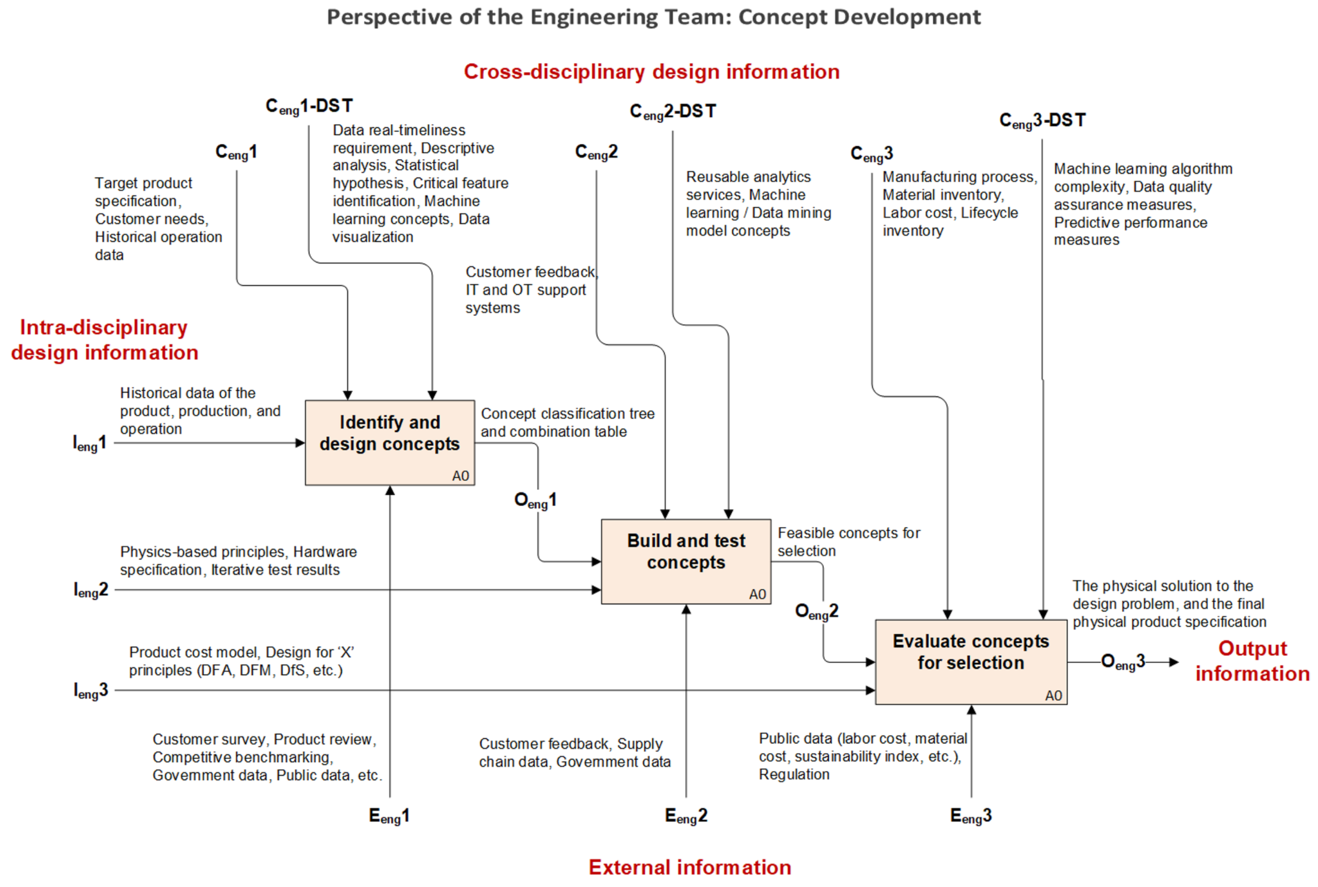

Perspective of the Engineering Team: Concept Development
Cross-disciplinary design information
C$_{eng}$1-DST
Data real-timeliness requirement, Descriptive analysis, Statistical hypothesis, Critical feature identification, Machine learning concepts, Data visualization
C$_{eng}$1
Target product specification, Customer needs, Historical operation data
C$_{eng}$2-DST
C$_{eng}$2
Reusable analytics services, Machine learning / Data mining model concepts
Customer feedback, IT and OT support systems
C$_{eng}$3
Manufacturing process, Material inventory, Labor cost, Lifecycle inventory
C$_{eng}$3-DST
Machine learning algorithm complexity, Data quality assurance measures, Predictive performance measures
Intra-disciplinary design information
I$_{eng}$1
Historical data of the product, production, and operation
Identify and design concepts
A0
Concept classification tree and combination table
O$_{eng}$1
Build and test concepts
A0
Feasible concepts for selection
O$_{eng}$2
I$_{eng}$2
Physics-based principles, Hardware specification, Iterative test results
Evaluate concepts for selection
A0
The physical solution to the design problem, and the final physical product specification
O$_{eng}$3
Output information
I$_{eng}$3
Product cost model, Design for 'X' principles (DFA, DFM, DfS, etc.)
Customer survey, Product review, Competitive benchmarking, Government data, Public data, etc.
Customer feedback, Supply chain data, Government data
Public data (labor cost, material cost, sustainability index, etc.), Regulation
E$_{eng}$1
E$_{eng}$2
E$_{eng}$3
External information

Fig. 8 Information flow from the engineers' perspective

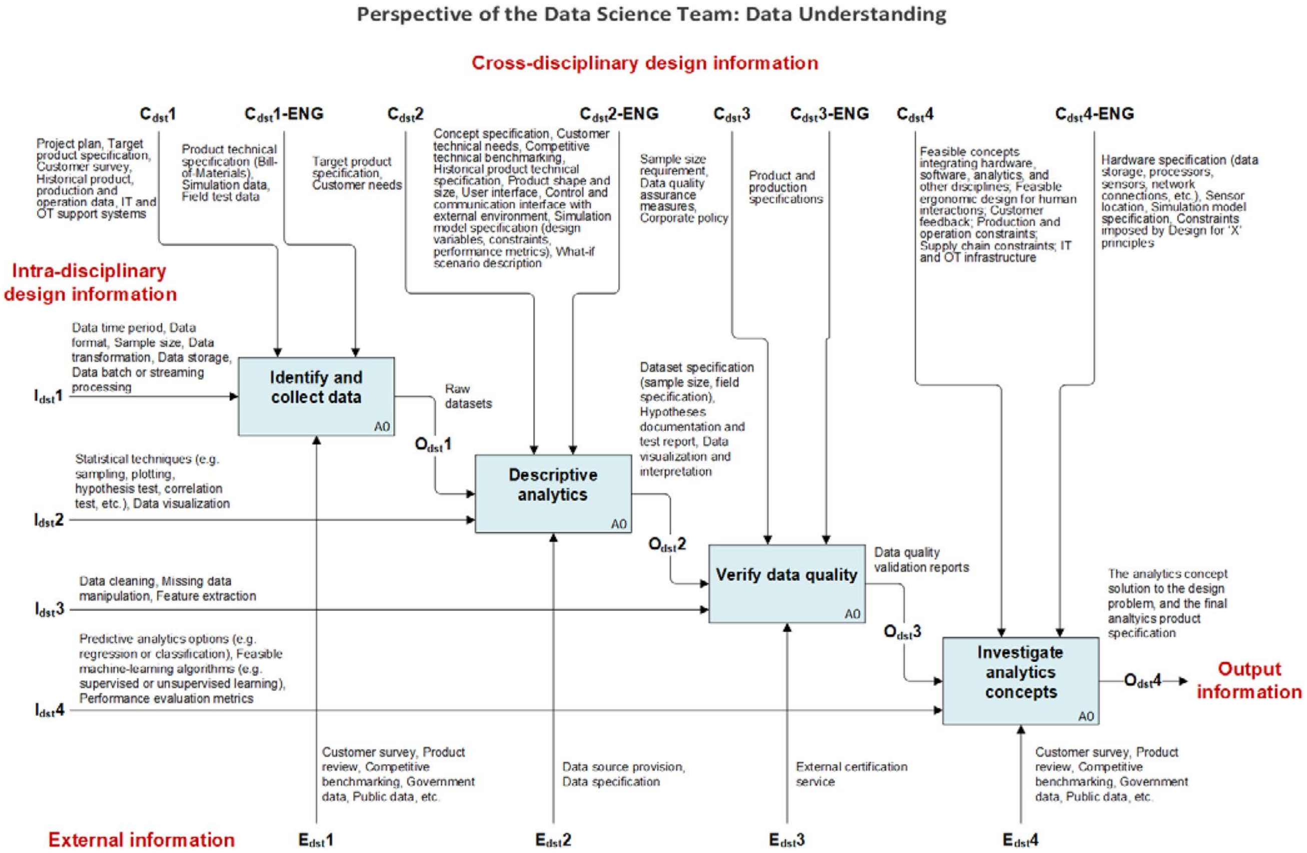

Perspective of the Data Science Team: Data Understanding
Cross-disciplinary design information
$C_{dst}1$
$C_{dst}1$-ENG
$C_{dst}2$
$C_{dst}2$-ENG
$C_{dst}3$
$C_{dst}3$-ENG
$C_{dst}4$
$C_{dst}4$-ENG
Project plan, Target product specification, Customer survey, Historical product, production and operation data, IT and OT support systems
Product technical specification (Bill-of-Materials), Simulation data, Field test data
Target product specification, Customer needs
Concept specification, Customer technical needs, Competitive technical benchmarking, Historical product technical specification, Product shape and size, User interface, Control and communication interface with external environment, Simulation model specification (design variables, constraints, performance metrics), What-if scenario description
Sample size requirement, Data quality assurance measures, Corporate policy
Product and production specifications
Feasible concepts integrating hardware, software, analytics, and other disciplines; Feasible ergonomic design for human interactions; Customer feedback; Production and operation constraints; Supply chain constraints; IT and OT infrastructure
Hardware specification (data storage, processors, sensors, network connections, etc.), Sensor location, Simulation model specification, Constraints imposed by Design for 'X' principles
Intra-disciplinary design information
Data time period, Data format, Sample size, Data transformation, Data storage, Data batch or streaming processing
$I_{dst}1$
Identify and collect data
A0
Raw datasets
$O_{dst}1$
Statistical techniques (e.g. sampling, plotting, hypothesis test, correlation test, etc.), Data visualization
$I_{dst}2$
Descriptive analytics
A0
Dataset specification (sample size, field specification), Hypotheses documentation and test report, Data visualization and interpretation
$O_{dst}2$
Data cleaning, Missing data manipulation, Feature extraction
$I_{dst}3$
Verify data quality
A0
Data quality validation reports
$O_{dst}3$
Predictive analytics options (e.g. regression or classification), Feasible machine-learning algorithms (e.g. supervised or unsupervised learning), Performance evaluation metrics
$I_{dst}4$
Investigate analytics concepts
A0
The analytics concept solution to the design problem, and the final analtyics product specification
$O_{dst}4$
Output information
Customer survey, Product review, Competitive benchmarking, Government data, Public data, etc.
Data source provision, Data specification
External certification service
Customer survey, Product review, Competitive benchmarking, Government data, Public data, etc.
External information
$E_{dst}1$
$E_{dst}2$
$E_{dst}3$
$E_{dst}4$

Fig. 9 Information flow from the data scientists' perspective

a dependency between these two concept generation processes. Hence, simulation with appropriate assumptions becomes a critical method to synchronize these two processes. This is consistent with the previous finding regarding the development of a data-driven manufacturing system (Li and Roy 2015).

Finally, Figs. 4, 8, and 9 together provide a more complete view of the NPD^3 model. The NPD^3 process model framework provides a starting point to understand how engineers and data scientists collaborate when they co-develop physical components and data-driven features involved in smart products. This understanding would also be beneficial to the integrated decision-making process design, which in return facilitates better product architecture design, for the development of a smart product.

7 Conclusion

A smart product can adapt itself to the environment where it is deployed and the data generated from its day-to-day use in turn improves its intelligence as well as benefits to all other instances. Creating these smart products requires developing two components—physical products (for physical bodies) and data products (for intelligence)—in a transdisciplinary approach across mechatronics, software, data science, and services domains. Specifically, mechanical and electrical engineers need to work closely with software engineers and data scientists to decide how to design the product to support more data-driven features. However, the misalignment of the product architecture and the development team organization may have a negative impact on product performance (Sosa et al. 2004). This is due to the fact that product-related interdependencies may not be addressed by the team's interactions, or that the design teams may interact in spite of the absence of a product-related interdependency.

To address our research key questions raised in Sect. 2 (which tasks need to be coordinated across the two team groups; when and what information needs to be exchanged between the two groups to collectively achieve the product development; and what are the patterns and characteristics of their interactions), this paper proposes NPD^3, an integrated process model for new product development with data-driven features. We revisited the classic NPD process model and a well-adopted data analytics process model, CRISP-DM, to understand the key tasks prescribed for the engineers in a physical product development team and the data scientists in a data product development team, respectively.

The NPD^3 model was then evaluated within a case study of the creation of a smart unmanned aircraft system. The results of our case study demonstrate that there was cross-disciplinary design information required by the engineers as well as the data scientists, and that it was critical that direct interactions and messages were exchanged between the two groups, with a project management group acting as a mediator to guide the collaboration across the team. In addition, the project management group helped to ensure that the required external design information was shared between the two disciplinary groups.

The timing and contents of information exchanged between the two groups facilitates information awareness and access, knowledge transfer, and problem-solving. We used four attributes—the number of interactions, the speed of information, the amount of contradictory information, and the uncertainty of the information—to characterize the interaction patterns. At the beginning of our project, a significant number of interactions were needed to identify the data sources and interests. As the process progressed, more data and information were available and the design problem was more constrained; therefore, the problem became less uncertain. When it came to the integration and evaluation of the concepts, the contradictory information increased because there had to be a compromise for the multiple competing performance measures. It was also noted that the speed of information decreased at first but then increased once the simulation model started to generate data based on various trial settings.

Our integrated process model is encoded with BPMN notation so that it can be implemented for automation, in a business management system, e.g. a PLM system. The NPD^3 model and the PLM implementation provided the UAS development team a collaborative environment and data repository to facilitate effective data/information exchange, visual communication, and traceable decision-making. The integrated process model also provided a common language and guidance to both engineers and data scientists, who otherwise would not have been familiar with the process used by their counterparts.

This work is a starting point to understand how engineers and data scientists should collaborate when they collectively need to develop future smart products that are highly data driven. We note that this UAS case study was carried out by university researchers and staff, and that a case study within an industry context might yield different results. However, we also note that UAS, as a tool or service, has been adopted by research scientists as well as within commercial applications, across a diverse and broad set of areas, including environment monitoring, infrastructure inspection, and precision agriculture (Gupta et al. 2013). The method used for within this case study is likely to be applicable to research scientists, as well as industry practitioners, who need to leverage UAS capabilities.

As with many empirical studies, however, the generality of our findings could be enhanced by conducting additional case studies for other smart products in different industries (Gibbert and Ruigrok 2010). We also realize the descriptive nature of the interaction patterns between engineers and data scientists. Hence, as more case studies are conducted, these patterns and characteristics could be further analyzed for quantitative evaluation and comparison. Last but not least, we only focused on the concept development stage in this paper; similar analyses could be conducted on other stages (e.g. sub-system design and detail design) of the product development process.

Acknowledgements This work has been partially funded by the Center for Advanced Systems and Engineering (CASE) at Syracuse University and the U.S. National Institute of Standards and Technology (NIST). The authors also sincerely appreciate all the Smart UAS project team members for providing the data and examples used in this paper.

References

Austin S, Baldwin A, Li B, Waskett P (1999) Analytical design planning technique: a model of the detailed building design process. Des Stud 20(3):279–296. https://doi.org/10.1016/S0142-694X(98)00038-6

Collins ST, Yassine AA, Borgatti SP (2008) Evaluating product development systems using network analysis. Syst Eng 12(1):55–68. https://doi.org/10.1002/sys.20108

Conforto EC, Amaral DC (2016) Agile project management and stage-gate model—a hybrid framework for technology-based companies. J Eng Tech Manage 40:1–14. https://doi.org/10.1016/j.jengtecman.2016.02.003

Cooper RG (1994) Perspective: third-generation new product processes. J Prod Innov Manag 11(1):3–14. https://doi.org/10.1016/0737-6782(94)90115-5

Cooper RG (2008) Perspective: the stage-gate idea-to-launch process—update, what's new, and NexGen systems. J Prod Innov Manag 25(3):213–232. https://doi.org/10.1111/j.1540-5885.2008.00296.x

Cooper RG (2014) What's next? After stage-gate. Res Technol Manag 57(1):20–31. https://doi.org/10.5437/08956308X5606963

Cooper RG (2016) Agile-stage-gate hybrids. Res Technol Manag 59(1):21–29. https://doi.org/10.1080/08956308.2016.1117317

Debevoise T, Taylor J (2014) The MicroGuide to process and decision modeling in BPMN/DMN: building more effective processes by integrating process modeling with decision modeling, CreateSpace independent publishing platform. ISBN: 978-1-502-78964-8

Dhar V (2013) Data science and prediction. Commun ACM 56(12):64–73. https://doi.org/10.1145/2500499

Distanont A, Haapasalo H, Kamolvej T, Meeampol S (2012) interaction patterns in collaborative product development (CPD). Int J Synergy Res 1(2):21–43

Eklund U, Bosch J (2012) Applying agile development in mass-produced embedded systems. In: Proceedings of international conference on agile software development (XP 2012), Malmö, Sweden, pp. 31–46. https://doi.org/10.1007/978-3-642-30350-0_3

Eklund U, Holmström OH, Strøm NJ (2014) Industrial challenges of scaling agile in mass-produced embedded systems. In:

Proceedings of international conference on agile software development (XP 2014), Rome, Italy, pp. 30–42. https://doi.org/10.1007/978-3-319-14358-3_4

Geiger C, Sarakakis G (2016) Data driven design for reliability. In: Proceedings 2016 annual reliability and maintainability symposium (RAMS), Tucson, AZ. https://doi.org/10.1109/RAMS.2016.7448023

Gericke K, Blessing L (2011) Comparisons of design methodologies and process models across disciplines: a literature review. In: Proceedings on 18th international conference on engineering design (ICED 11), Lyngby/Copenhagen, Denmark, 1, pp. 393–404

Gerwin D, Barrowman NJ (2002) An evaluation of research on integrated product development. Manage Sci 48(7):938–953. https://doi.org/10.1287/mnsc.48.7.938.2818

Gibbert M, Ruigrok W (2010) The 'What' and 'How' of case study rigor: three strategies based on published work. Organ Res Methods 13(4):710–737. https://doi.org/10.1177/1094428109351319

Gupta SG, Ghonge MM, Jawandhiya PM (2013) Review of unmanned aircraft system (UAS). Int J Adv Res Comput Eng Technol (IJARCET) 2(4):1646–1658

Herrmann JW, Schmidt LC (2002) Viewing product development as a decision production system. In: Proceeding of ASME 2002 international design engineering technical conferences and computers and information in engineering conference (DETC2002), Montreal, Canada, 4, pp. 323–332. https://doi.org/10.1115/DETC2002/DTM-34030

Huang HM (2008) Autonomy levels for unmanned systems (ALFUS) framework volume i: terminology. Special Publication, National Institute of Standards and Technology (NIST), Gaithersburg, MD. 1011-I-2.0

Karlström D, Runeson P (2005) Combining agile methods with stage-gate project management. J IEEE Softw 22(3):43–49. https://doi.org/10.1109/MS.2005.59

Karlström D, Runeson P (2006) Integrating agile software development into stage-gate managed product development. Empir Softw Eng 11(2):203–225. https://doi.org/10.1007/s10664-006-6402-8

Kassner L, Gröger C, Mitschang B, Westkämper E (2015) Product life cycle analytics—next generation data analytics on structured and unstructured data. In: Proceedings on 9th CIRP conference on intelligent computation in manufacturing engineering (CIRP ICME'14), Naples, Italy, 33, pp. 35–40. https://doi.org/10.1016/j.procir.2015.06.008

Kendoul F (2012) Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. J Field Robot 29(2):315–378. https://doi.org/10.1002/rob.20414

Krishnan V, Ulrich KT (2011) Product development decisions: a review of the literature. Manage Sci 47(1):1–21. https://doi.org/10.1287/mnsc.47.1.1.10668

Krovi R, Chandra A, Rajagopalan B (2003) Information flow parameters for managing organizational processes. Commun ACM 46(2):77–82. https://doi.org/10.1145/606272.606277

Kurgan LA, Musilek P (2006) A survey of knowledge discovery and data mining process models. Knowl Eng Rev 21(1):1–24. https://doi.org/10.1017/S0269888906000737

Li Y, Roy U (2015) Challenges in developing a computational platform to integrate data analytics with simulation-based optimization. In: Proceedings on ASME 2015 international design engineering technical conference and computers and information in engineering conference (DETC2015), Boston, MA, pp. V01BT02A035. https://doi.org/10.1115/DETC2015-46410

Li Y, Roy U (2018) A multi-objective optimization methodology towards product design for sustainability. Int J Strateg Eng Asset Manag 3(2):154–176. https://doi.org/10.1504/IJSEAM.2018.092234

Li Y, Roy U, Shin SJ, Lee YT (2015) A 'Smart Component' data model in PLM. In: Proceedings on 2015 IEEE international conference on big data, Santa Clara, CA, pp. 1388–1397. https://doi.org/10.1109/BigData.2015.7363899

Li Y, Roy U, Saltz JS (2017) Modular design of data-driven analytics models in smart-product development. In: Proceedings on ASME 2017 international mechanical engineering congress & exposition (IMECE2017), Tampa, FL, USA. pp. V011T15A022. https://doi.org/10.1115/IMECE2017-71597

Lohr S (2011) Ex-apple leaders push the humble thermostat into the digital age. The New York Times, http://www.nytimes.com/2011/10/25/technology/at-nest-labs-ex-apple-leaders-remake-the-thermostat.html. Accessed 20 Oct 2018

Marbán O, Segovia J, Menasalvas E, Fernández-Baizán C (2009) Toward data mining engineering: a software engineering approach. Inf Syst 34(1):87–107. https://doi.org/10.1016/j.is.2008.04.003

NATO (2012) Standard interfaces of UAV control system (UCS) for NATO UAV interoperability, 3rd ed., NATO Standardization Agency (NSA). STANAG 4586

Nest Labs (2012) Nest learning thermostat efficiency simulation: update using data from first three months. White Paper, Nest, Palo Alto, CA. http://downloads.nest.com/efficiency_simulation_white_paper.pdf. Accessed 20 Oct 2018

Nest Labs (2013) The effect of schedules on HVAC runtime for nest learning thermostat users. White Paper, Nest, Palo Alto, CA

Nest Labs (2014) Enhanced auto-schedule. White Paper, Nest, Palo Alto, CA. https://nest.com/downloads/press/documents/enhanced-auto-schedule-white-paper.pdf. Accessed 20 Oct 2018

Ore JP, Elbaum S, Burgin A, Detweiler C (2015) Autonomous aerial water sampling. J Field Robot 32(8):1095–1113. https://doi.org/10.1002/rob.21591

Patil DJ (2012) Data Jujitsu: the art of turning data into product. O'Reilly Media, Inc., CA. ISBN: 978-1-449-34115-2. http://radar.oreilly.com/2012/07/data-jujitsu.html. Accessed 20 Oct 2018

Pavliscak P (2015) Data-informed product design, 1st edn. O'Reilly Media, Inc., CA. ISBN: 978-1-491-93129-5. https://www.oreilly.com/ideas/data-informed-product-design. Accessed 20 Oct 2018

Peffer T, Pritoni M, Meier A, Aragon C, Perry D (2011) How people use thermostats in homes: a review. Build Environ 46(12):2529–2541. https://doi.org/10.1016/j.buildenv.2011.06.002

Porter ME, Heppelmann JE (2015) How smart, connected products are transforming companies. Harvard Bus Rev 93:96–114

Roy U, Zhu B, Li Y, Zhang H, Yaman O (2014) Mining big data in manufacturing: requirement analysis, tools and techniques. In: Proceedings on ASME 2014 international mechanical engineering congress and exposition (IMECE2014), Montreal, Canada, 11, pp. V011T14A047. https://doi.org/10.1115/IMECE2014-38822

Saltz JS (2015) The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness. In: Proceedings on 2015 IEEE international conference on big data, Santa Clara, CA, pp. 2066–2071. https://doi.org/10.1109/BigData.2015.7363988

Shearer C (2000) The CRISP-DM model: the new blueprint for data mining. J Data Warehous 5(4):13–22

Solomatine DP, Ostfeld A (2008) Data-driven modelling: some past experiences and new approaches. J Hydroinform 10(1):3–22. https://doi.org/10.2166/hydro.2008.015

Sosa ME, Eppinger SD, Rowles CM (2004) The misalignment of product architecture and organizational structure in complex product development. Manage Sci 50(12):1674–1689. https://doi.org/10.1287/mnsc.1040.0289

Sun K, Li Y, Roy U (2017) A PLM-based data analytics approach for improving product development lead time in an engineer-to-order manufacturing firm. Math Model Eng Probl 4(2):69–74. https://doi.org/10.18280/mmep.040201

Ulrich KT, Eppinger SD (2012) Product design and development, 5th edn. McGraw-Hill, New York. ISBN: 978-0-073-40477-6

Unger DW, Eppinger SD (2009) Comparing product development processes and managing risk. Int J Prod Dev. https://doi.org/10.1504/IJPD.2009.025253

Wheatcraft LS, Ryan MJ, Svensson C (2017) Integrated data as the foundation of systems engineering. In: 27th annual INCOSE international symposium (IS 2017), Adelaide, Australia, vol 27(1), pp. 1423–1437. https://doi.org/10.1002/j.2334-5837.2017.00438.x

Wynn DC, Clarkson PJ (2018) Process models in design and development. Res Eng Design 29(2):161–202. https://doi.org/10.1007/s00163-017-0262-7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.