



A data-driven cyber-physical approach for personalised smart, connected product co-development in a cloud-based environment

Pai Zheng^{1,2,3} · Xun Xu² · Chun-Hsien Chen³

Received: 26 November 2017 / Accepted: 15 June 2018 / Published online: 18 June 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

The rapid development of information and communication technology enables a promising market of information densely product, i.e. smart, connected product (SCP), and also changes the way of user–designer interaction in the product development process. For SCP, massive data generated by users drives its design innovation and somehow determines its final success. Nevertheless, most existing works only look at the new functionalities or values that are derived in the one-way communication by introducing novel data analytics methods. Few work discusses about an effective and systematic approach to enable individual user innovation in such context, i.e. co-development process, which sets the fundamental basis of the prevailing concept of data-driven design. Aiming to fill this gap, this paper proposes a generic data-driven cyber-physical approach for personalised SCP co-development in a cloud-based environment. A novel concept of smart, connected, open architecture product is hence introduced with a generic cyber-physical model established in a cloud-based environment, of which the interaction processes are enabled by co-development toolkits with smartness and connectedness. Both the personalized SCP modelling method and the establishment of its cyber-physical product model are described in details. To further demonstrate the proposed approach, a case study of a smart wearable device (i.e. i-BRE respiratory mask) development process is given with general discussions.

Keywords Data-driven design · Open architecture product · Smart product · Cyber physical system · Cloud · Mass personalisation

Introduction

The rapid development of information and communication technology (ICT) has caused products to evolve from usual products (i.e. mechanical or physical products) to smart products (i.e. usual products with embedded system), or information technology (IT)-enabled products (i.e. usual products with connectedness), and are converging to the so-called smart, connected products (SCP) (Porter and Hep-

pelmann 2014) (i.e. usual products with embedded ICT components), as shown in Fig. 1.

SCP, with embedded ICT components, have the abilities to collect, process, produce information and somehow to “think by itself” (Rijsdijk and Hultink 2009). For example, with a specific APP installed, a smart phone can be utilised for monitoring walking steps or informing upcoming events other than sending text message or ringing someone. Owing to the embedded open toolkits (Franke and Piller 2004), SCP provides a promising way for mass personalisation (Tseng et al. 2010; Zhou et al. 2013). Users instead of designers can create their own products through adaptable hardware interface or application programming interface (API), following a specific set of rules or guidance. For instance, users can make new applications, such as a temperature monitor based on Raspberry Pi, to generate customized solutions by virtual prototyping in an online product platform (Carulli et al. 2013). Conventionally, the detailed design information is either acquired via market research or assumed via professional knowledge, which is not effective and often results in

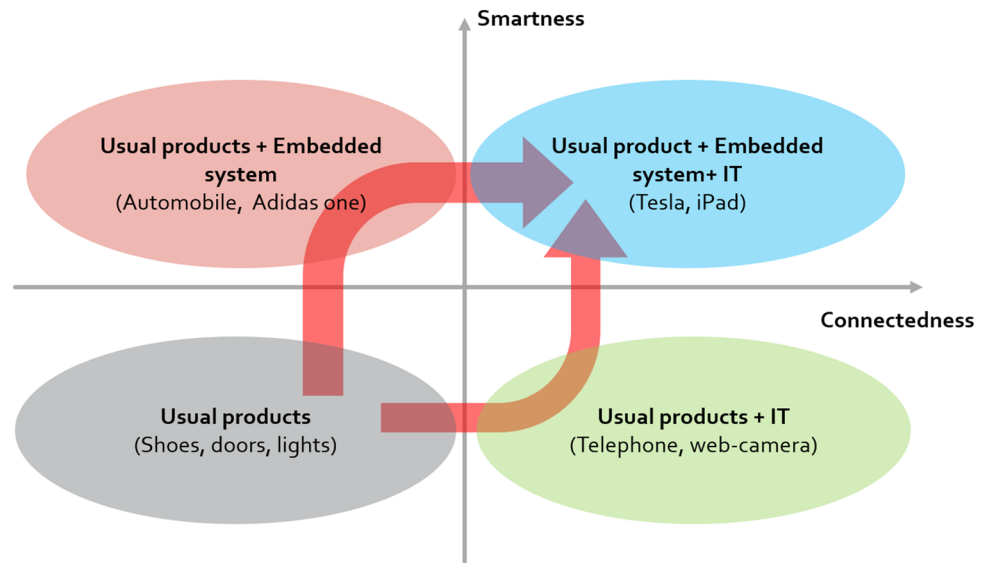
✉ Pai Zheng
ericzheng@ntu.edu.sg; pzhe539@aucklanduni.ac.nz

¹ Delta-NTU Corporate Laboratory for Cyber-Physical System, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

² Department of Mechanical Engineering, University of Auckland, Private Bag 92019, Auckland, New Zealand

³ School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore

Fig. 1 Product evolution towards smartness and connectedness



an ever-rising new product development flop rates (Henkel and Von Hippel 2004; Piller et al. 2010). However, for SCP, due to their connectivity, e.g. Internet of Things (IoT), companies have real time access to the massive user generated data, which enables the data-driven continuous design improvement (or “evergreen design”) and next-generation product prediction (Porter and Heppelmann 2015). Hence, these unique characteristics require the way of product development to be shifted to a data-driven manner so as to meet individual customer satisfaction.

Despite the above characteristics, to the authors’ knowledge, most existing works of SCP only look at the new functionalities or values that are derived from the user-side by introducing novel data analytical methods. Few works discuss about how to systematically and effectively enable its individual user innovation, i.e. personalized SCP development, which acts as the basis of data-driven design innovation. Aiming to fill this gap, this paper proposes a generic approach for personalised SCP co-development. The rest of this paper is organised as follows. “[Theoretical background](#)” section provides basic notions of SCP and an overview of related works. “[Enabling tools for SCP co-development user innovation](#)” section discusses enabling tools for SCP co-development and introduces a novel concept of smart connected open architecture product (SCOAP). “[Establishment of a cloud-based platform](#)” section establishes a data-driven cyber-physical model for personalized SCP modelling in the cloud-based environment. A case study of a personalized smart wearable development is given in “[Case study](#)” section. The novelty and potential applications of the proposed approach is discussed in “[Discussions](#)” section. At last, conclusions and future work are summarised.

Theoretical background

In order to have a better understanding of data-driven SCP development, this section summarizes the basic notions of SCP, and gives a comprehensive review of the recent development of data-driven design, and its co-development toolkits to enable such product co-development process.

Basic notions of SCP

SCP represents the third wave of IT-driven competition, with IT embedded in the products. It changes the way how value is created. Different from the definition of IoT, which is known as a mechanism for transmitting information by providing ubiquitous connectivity at a low cost, its fundamental value lies in the changing nature of the product itself (Porter and Heppelmann 2015).

According to Porter and Heppelmann (2014), SCPs mainly consist of three core elements: (1) *physical components* which comprise the electrical and mechanical parts as usual product’s; (2) *smart components* which comprise the sensors, microprocessors, data storage, controls, software, and, typically, an embedded operating system and enhanced user interface. In many cases, software allows a single physical device or replaces some hardware components to perform at multi-levels; and (3) *connectivity components* which comprise the ports, antennae, and protocols enabling wired or wireless connections with the product. It enables not only information exchange between the operating environment and the product, but also “product cloud” functions which exist outside the product. For example, iPhone, as a SCP, can be connected to the Internet through Wi-Fi, Bluetooth or 3G/4G so that each user can benefit from utilizing various applications from the APP store. The major features (capa-

bilities) of SCPs include (Porter and Heppelmann 2014): monitoring (e.g. real-time visibility of the product condition), control (e.g. operator controls a robot from iPad), optimisation (e.g. cutting tool wear prediction) and autonomy (e.g. auto robot vacuum cleaner).

Data-driven design

Data-driven design is a newly brought up term with the arrival of big data, IoT and Cyber-Physical Systems (CPS). Massive user/product generated data enables this new form of design methodology. Such informatics-based approach serves as the key for value creation by providing the ability to identify user behavior patterns or latent needs (Lim et al. 2018). Despite a lack of unified definition, data-driven design has been widely applied in many areas, such as: final product quality prediction (Wang 2011), industrial control system fault detection and diagnosis (Yin et al. 2014), predicting the remaining useful life of critical components (Mosallam et al. 2016), online learning courses design (Kizilcec and Schneider 2015), to name a few. Regarding the scope of this research, we only select the literature relevant to the product design stage, which can be further classified into the following three aspects.

Product design evaluation Tucker and Kim (2009) presented a decision tree based data mining technique to optimise the product portfolio design and selection process by eliminating the tremendous time and resource cost. Kusiak (2009) proposed a data-driven approach to evaluate the innovativeness of planned introductions of design changes and design of new products and services from a market perspective. Wang et al. (2011) developed a systematic data-mining approach for eliciting product attributes from the web-based user generated content, and constructed user preference model for design selection decision making.

Product design prediction Wang and Chen (2015) introduced a data-driven network analysis method (i.e. product association network) to predict personalised choice sets for engineering design decision-making based on existing data of customer choice sets. Tuarob and Tucker (2015) provided a text mining knowledge-based system that identifies product features and associated customer preferences from the source of large scale, heterogeneous social media data in the market, and integrated them into a new product design. Ma and Kim (2016) proposed a predictive data-driven model for determining the optimal product family architectures with customer preference data by k-means clustering.

User experience-based design innovation Lin et al. (2016) proposed a six-phase UNISON framework for data-driven innovation to capture user experience and preference of product form designs to enhance customer satisfaction. Chien

et al. (2016) also presented a data-driven design framework for extracting user experience of product visual aesthetics to identify useful design concepts in regards of user preference and user response. Gupta et al. (2018) introduced a systematic approach for gathering requirements (i.e. user feedback) during aircraft production by adopting modular architecture and augmented reality approach in a remote manner where user in-context information can be retrieved.

Co-development toolkits

To facilitate the personalised product development, several co-development toolkits have been proposed and developed in literature. Generally, they can be classified into two categories, i.e. product configuration toolkit (Franke and Piller 2002) and embedded open toolkit (Piller et al. 2010).

Configuration toolkit

Configuration toolkit, also referred to as product configuration system (PCS) or mass customisation toolkit (Franke and Piller 2002), is a knowledge-based system to tailor a product according to the specific needs of a customer (Jannach and Zanker 2013) with a shorter lead time to market (Salvador and Forza 2004). It consists of a set of predefined attributes with constraints (rules) for customer to select within the product family scope (Xie et al. 2005). In the configuration process, the input is the customer's selection of existing attributes and the output is the recommended or target product derived from the system to fulfil customer requirements (CRs). In such a way, it bridges the gap between CRs and the end-product by only a series of attribute selection processes in a "configure-to-order" (CTO) model. Moreover, it benefits the company by reusing existing design elements to provide customer-perceived product variety in the product family (Wang and Tseng 2014; Chen and Wang 2008).

Trentin et al. (2012) summarised the fundamental functions of a PCS as: (1) communicating company's product offerings to the customer; (2) providing real-time information, such as quotation, delivery time and product specification; (3) checking the completeness and validity of product variant; and (4) generating bill-of-materials based on customer's selection. Except for these functions, the traditional centralised, CTO-based configuration toolkit is moving towards decentralised (Zhang 2014), engineering to order (ETO)-based model (Levandowski et al. 2015) with intelligent methods (Wang and Tseng 2014) to meet the ever increasing personalised CRs and to empower more users' flexibility in the co-development process.

Embedded open toolkit

Embedded open toolkit, also known as product with built-in flexibility, was first proposed by Franke and Piller (2004) as a scalable attempt towards “open hardware”, i.e. extending the open innovation paradigm towards tangible products, by embedding knowledge and rules about possible product differentiations into the product (Piller et al. 2010). Other than the configuration toolkit, which only allows users to select or combine design modules and parameters in the design stage, the open toolkit is embedded in the product architecture, allowing real-time modification during the usage stage along its life-cycle via adaptable interfaces. It is claimed as a postponement method in new product development to increase design flexibility (Gross and Antons 2009), which is built on shifting some specifications of the product into the customer domain after the product manufactured (Piller and Walcher 2006). According to Piller et al. (2010), Gross and Antons (2009), an embedded open toolkit should contain: (1) a flexible architecture where the values for each design parameter are not fixed, but adaptable, (2) a set of rules about possible combinations of selection of values for each design parameter, and (3) an interface for individual users could differentiate the product according to their preferences by manipulating the values. In such a way, users are enabled to adapt a physical product directly based on their own needs.

With the arrival of SCP, the concept of embedded open toolkit is broadened to the integration of open software innovation (e.g. APPs in smart phone) and open hardware innovation (e.g. Raspberry Pi) as a whole, which is known as smart product with built-in flexibility (Bénade et al. 2016). Due to the ever-increasing flexibility, it has the unique characteristics that users, instead of manufacturers, endorse the role of designing their own uses under certain guidance (Brown 2013; Bénade et al. 2016).

Enabling tools for SCP co-development user innovation

Figure 2 depicts the evolvement of product co-development toolkits in correspondence with the product evolvement paradigm depicted in Fig. 1. Originally, usual product development is enabled by offline marketing strategies (e.g. user feedback, focus group). With the development of embedded technology and IT, embedded open toolkits (e.g. smart sensors) and online software toolkits (e.g. product configuration system) have been widely utilized, respectively. The prevailing integration of ICT technologies eventually guarantees the SCP with embedded smart open toolkits in a connected environment.

In order to discover a systematic and effective way to enable user data acquisition and user innovation, this section

discusses a potential method by integrating the two prevailing co-development toolkits, i.e. online configuration toolkit and embedded open toolkits in an open environment for personalized SCP co-development. Then, based on it, a novel concept of SCOAP is introduced.

Integration of co-development toolkits

The essence of data-driven SCP co-development can be seen as a new form of product design innovation. It is beyond the pragmatic view of systematic design theories [e.g. axiomatic design (Suh 1990), function-behaviour-structure design (Gero 1990)], as a dynamic mapping process between the function domain and physical domain, but modelled as an interplay between the *design solution space (DSS)*, and the *user solution space (USS)*, as shown in Fig. 3.

Following this manner, the SCP co-development process can be further explained as:

- $DSS \rightarrow USS$: a manufacturer’s defined solution space drives a user-generated solution, for example, a customized bicycle by undertaking online configuration.
- $USS \rightarrow DSS$: a user-generated solution triggers the innovation of a manufacturer’s design solution, for example, the real-time monitoring of machine tools usage by users motivating the new product.
- $DSS \rightarrow DSS$: a manufacturer’s design solution urges self- or competitor-design innovation, for example, the introduction of fingerprint identification prompts a widespread application of smart phones.
- $USS \rightarrow USS$: one or more user-generated solutions contribute to other users’ innovations, for example, open source plugins contributed by one user can be edited by others, based on their own needs.

The co-development interactions between users and designers can be conducted either in the product prototype development process before the end-product is manufactured by utilizing PCS (Zheng et al. 2017a), or after manufactured in the customer domain of usage with embedded open toolkits. Both spaces are expanded jointly through the action of design operators, viz. *design iteration*, and the total space of design solution and user solution is defined as *product attainable solution space*, as shown in Fig. 4.

For PCS, it includes a set of pre-defined product attributes with configurable components (CCs), which users can tailor their preferred designs within the product family scope before manufactured, as shown in Fig. 4a. This type of co-development model is defined as *Smart product with CCs*. The manufacturer rather than the user drives the design innovation, and users usually select from existing options by utilizing the online product configuration toolkit before it is manufactured (i.e. $DSS \rightarrow USS$). Once determined, the

Fig. 2 Evolution of product development toolkits towards smartness and connectedness

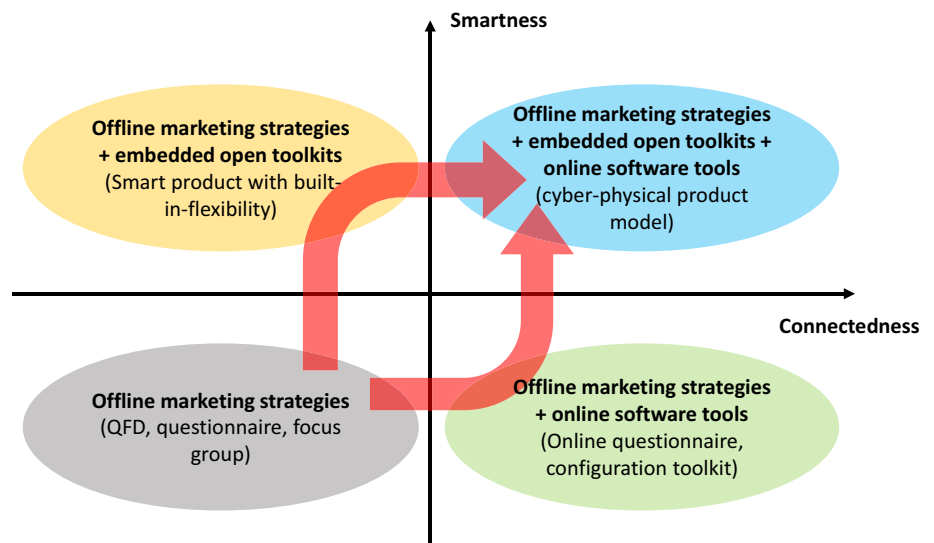
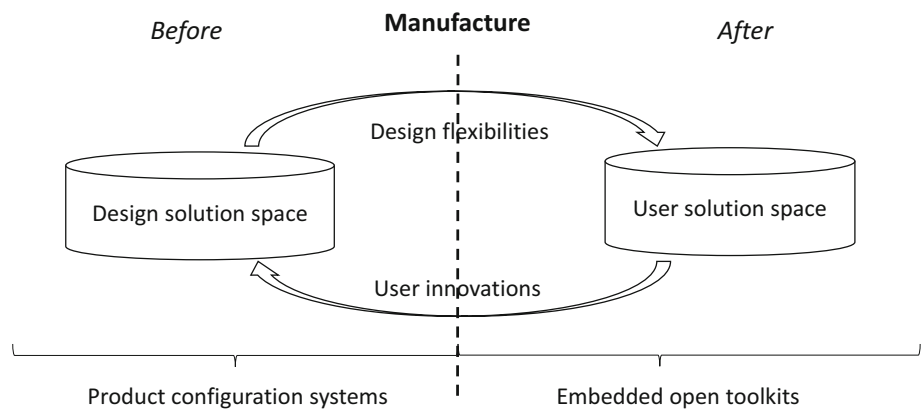


Fig. 3 Integration of co-development toolkits for personalized SCP innovation



end-product cannot be adjusted continuously in its life-cycle usage. For example, users can only change the colour and type of the Fitbit wristband offered by the company.

For embedded open toolkit, it allows certain degree of customer design freedom by embedding knowledge and rules about possible product differentiations into the product after manufactured, as shown in Fig. 4b. This type of co-development model is defined as *Smart product with built-in-flexibility*. The end-product can be adjusted continuously in its life-cycle usage. Nevertheless, it still correlates much with the original product offered by the manufacturer and user generated data are often confined in the scalable usage options (i.e. $USS \rightarrow DSS$, $DSS \rightarrow USS$). For example, Adidas One is a kind of smart running shoes that users can change its modes, based on the usage context.

One can find that neither of them alone can fully allow user innovation along the whole product co-development process, and it is claimed that they should be integrated together, with an open environment (i.e. $DSS \rightarrow DSS$ and $USS \rightarrow USS$) to achieve a higher degree of open hardware and software innovation flexibility. This new type of co-development is named,

SCOAP, as shown in Fig. 4c. In *SCOAP*, *user solution space* normally becomes much larger than *design solution space*. For instance, Raspberry Pi, has been exploited in many applications by various users far beyond its original educational purpose.

Smart connected open architecture product

Inspired by the concept of open architecture product introduced by Koren et al. (2013), the concept of *SCOAP* is proposed and defined as:

an IT-driven product consisting of physical, smart and connectivity components, with a platform containing both open hardware and software toolkits, that allows the integration of modules from different sources in order to adapt product and its service functionality exactly to the user's own needs throughout lifecycle.

By this definition, *SCOAP* follows the adaptable design principles (Gu et al. 2009), which aim to deliver designs and products fulfilling ever-changing CRs with an extended prod-

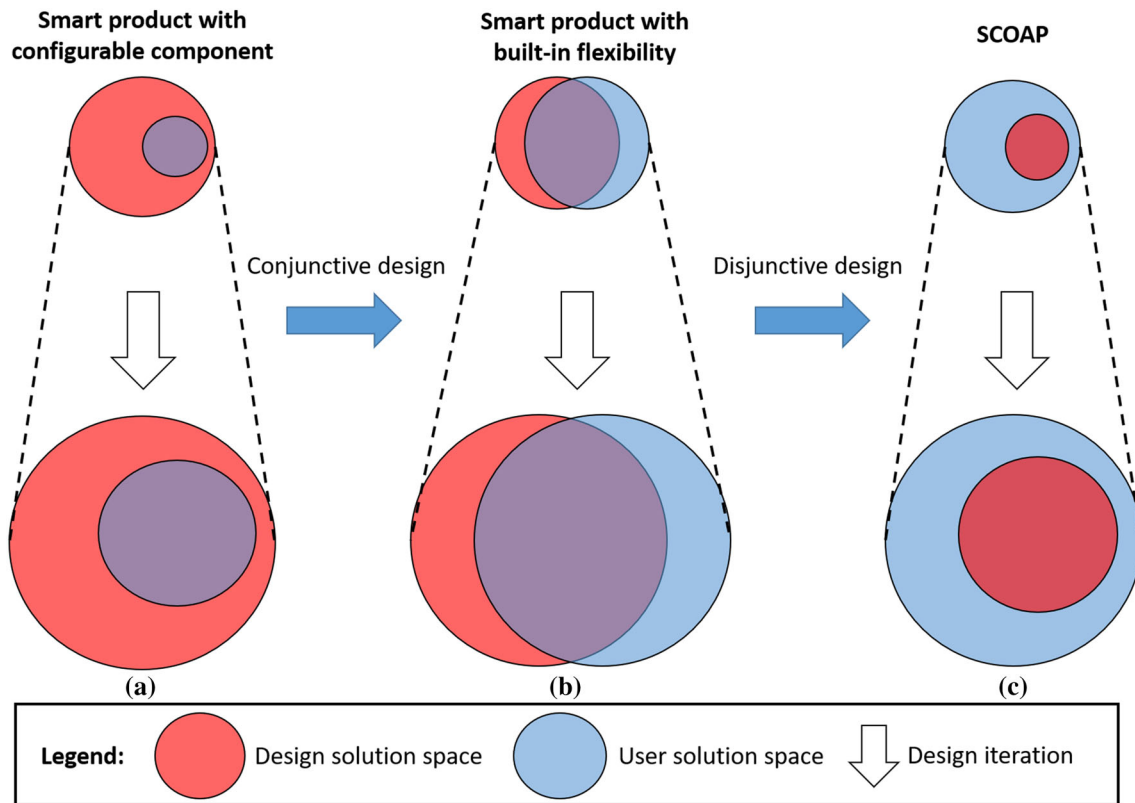


Fig. 4 Typical types of co-development models and their interrelationship between DSS and USS for SCP design innovation

uct lifecycle and represents nowadays product development towards highly smartness and connectedness. Moreover, it enlarges the scope of open architecture product as an open hardware innovation approach, by integrating open software innovation as well. Considering the open innovation environment, with many companies or suppliers joining the platform, each one can cooperate with many others by delivering or outsourcing part of its design resources or workload. Meanwhile, one can achieve abundant design resources and requirement information on demand, which again enables the ‘market-of-one’.

Establishment of a cloud-based platform

Following the definition of SCOAP, one key issue is to establish a proper platform in an open environment to enable user data acquisition and user innovation. With the prevailing development and application of cloud computing and service oriented architecture, cloud-based environment has been widely adopted in the manufacturing area (i.e. cloud manufacturing) (Xu 2012). It has shown unique advantages in enabling distributed product development process, including a decreased time span, effective information communication, and abundant resources in security (Zheng et al. 2017b). Furthermore, by adopting a cloud-based approach, companies

do not need to invest much money in the IT infrastructure. Instead, they can have instant access to business and technical solutions in a ‘pay-per-use’ manner in the cloud. Owing to its prominent advantages and also inspired by the concepts of CPS (Lee 2008), this work proposes a data-driven cyber-physical approach in a cloud-based platform for SCP co-development process, as shown in Fig. 5. Its information communication process, SCP modelling process, and cyber-physical product model are described as follows.

Information communication

As summarised by Yu and Zhu (2016), there are two dimensions of data-driven product design: the market and user-generated data for product development (e.g. user feedback), and the data serving as part of the product value (e.g. heart-rate monitoring). In a data-driven personalised SCP development, both are essential to enabling customer satisfaction and achieved by the effective information communication among three key roles, i.e. user, product, and manufacturer.

In the cyber space, i.e. cloud-based environment, cloud-users, acting as the cloud-service demander requests a personalized product by interacting with the SCOAP platform through graphical user interfaces (GUIs). The cloud-users

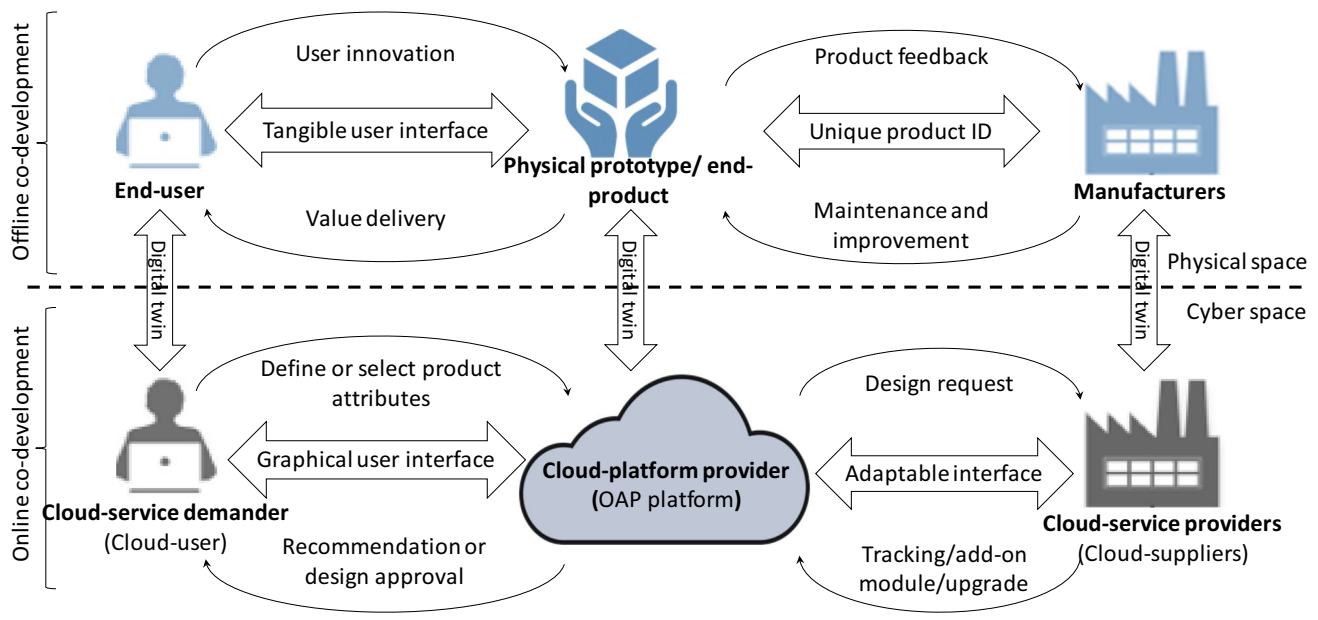


Fig. 5 A proposed cloud-based data-driven cyber-physical approach for personalized SCP co-development

can either conduct a product configuration process to select a customized product attribute existed on the product platform or submit a new request to the cloud platform. Meanwhile, the SCOAP platform, hosted by the cloud-platform provider, should capture useful user information either through embedded ICT components, or from user's online input information. Also, a knowledge base should be established including all the design constraints of the product family. Meanwhile, the SCOAP platform provides adaptable interfaces for the multiple suppliers, i.e. cloud-service providers, to upload add-on modules or upgrade their online design resources under specific rules (design constraints). Reversely, the cloud-service providers can track real-time information of their modules from the embedded ICT components by login to the SCOAP platform. Once a new design request is beyond the existing product family scope, the SCOAP should deliver it to the suitable cloud-service providers for their new design and approving/declining intelligently. Due to its openness, users can share their designs in the online communities to motivate the user generated design innovation (*i.e.* $USS \rightarrow USS$), and suppliers can compete or outsource part of its design workload with others to enable the manufacturer generated design innovation (*i.e.* $DSS \rightarrow DSS$).

Meanwhile, in the physical space, end-users have the tangible experience of the prototype/end-product and can make changes based on their own preferences and the constraints of the embedded open toolkit. The personalized SCP, delivers the value to achieve customer satisfaction, and also tracks important information of the user (e.g. location information, heart rate). The information will be sent to the manufacturers as customer feedback for improving their existing design or

for product maintenance purposes. The physical space also provides an offline co-development environment including face-to-face communication, interviews, live maintenance, etc.

Moreover, the cyber space and physical space are interconnected as the digital twin, which the end-user and manufacturer will have a unique online identification in the cloud (e.g. username) and other essential information to guarantee the privacy and security, while the SCP digital twin requires more information, as the cyber model should even fully reflect the physical model if needed.

SCOAP modelling

In the proposed approach, the establishment of a SCOAP is critical, which enables the product flexibility for user innovation. Therefore, an appropriate product modelling methodology should be defined as the backbone. In this research, a two-stage based product modelling method (*i.e.* modular design stage and scalable design stage) is proposed (Fig. 6). The modular design is at a macro-level focusing on the performance of the entire product family design, while the scalable design is at a micro-level concentrating on the optimisation of design specifications (Du et al. 2014). They both guarantee the flexibility and openness of the SCP in a bi-level manner.

Moreover, considering the unique characteristics of SCP, SCOAP modelling is classified into three levels: *physical product level*, *embedded ICT hardware level* and *embedded ICT software level*, respectively. Each subsequent level is modelled as an extension of the current one. For example,

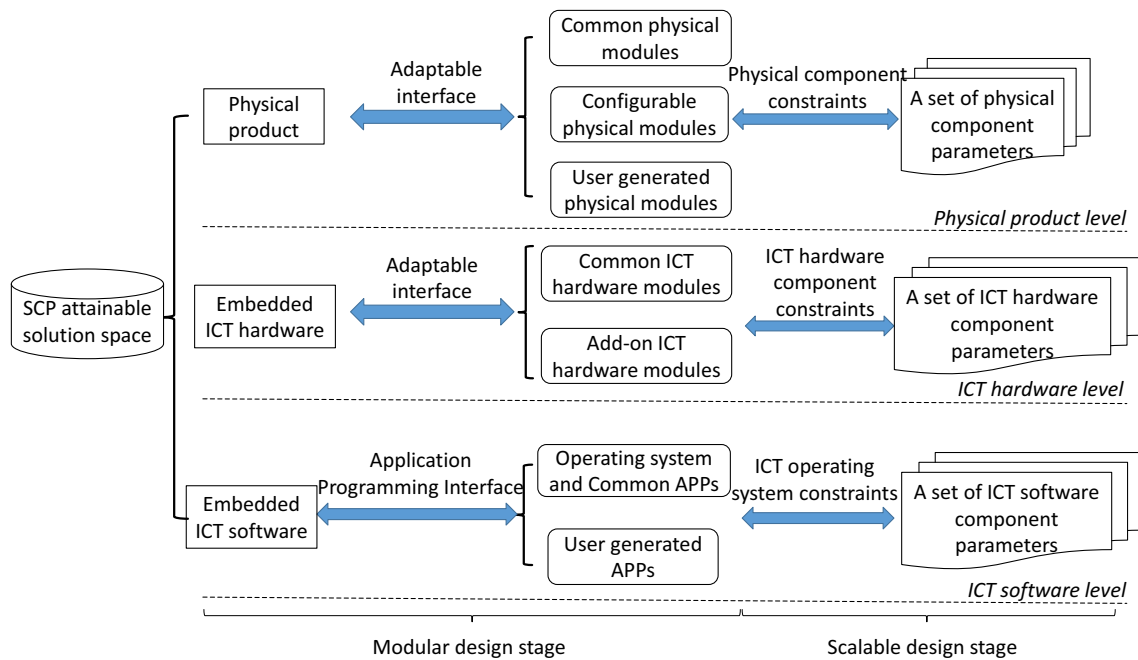


Fig. 6 A proposed two-stage design process for SCOAP modelling

counting walking steps is an extended smart function application based on physical running shoes. Also, the degree of interdependency between the former level and the latter determines the type of data-driven co-development process, that is, *conjunctive design* or *disjunctive design*. For example, there lies a large interdependency between the physical product level, and embedded hardware and software levels of Adidas One running shoes, so that users cannot generate much innovation beyond the original SCP (*conjunctive design*). In contrast, iPhone, for instance, has no obvious interdependency between each level, which enables users to create many APPs far beyond the basic function of calling service (*disjunctive design*). The details of the proposed two-stage design process of three-level product modelling are described below.

Modular design of SCP

At the modular design stage, one should follow the principles of adaptable design (Gu et al. 2004), which stands for the ability of a design or a product to be adapted to new requirements and be reused when circumstances change, by adding or replacing certain modules through a pre-defined adaptable interface. Functional modelling is the key enabler for each design module to be independent of the others, so that the SCP can be easily upgraded or changed based on individual user needs. Meanwhile, the adaptable interfaces along with the design rules/constraints should be standardized by the platform provider to enable the user evaluation and design consistency by different stakeholders in a guided manner.

Physical product mainly consists of three types of modules: *common physical modules*, *configurable physical modules*, and *user-generated physical modules*. *Common physical modules* are the unchangeable physical modules, for example, the engine of an automobile. *Configurable physical modules* have a set of CCs, for example, gear box of an automobile. Both are pre-defined by the manufacturer. *User-generated physical modules* are add-on modules that can be designed by users individually through an adaptable interface, for example, a navigation module can be added to an automobile by inserting it into the slot.

Embedded ICT hardware mainly consists of *common ICT hardware modules* and *add-on ICT hardware modules*. Normally, the ICT hardware remains unchanged and sometimes can be replaced by upgrading the software. However, in order to be flexible enough for add-on functions, the hardware design should still consider *add-on ICT hardware modules* enabled by adaptable interfaces (i.e. an open hardware toolkit). For example, the memory size of an Android device can be expanded by adding memory stick.

Embedded ICT software mainly consists of *operating system with common APPs*, that is, those pre-embedded in the hardware by the manufacturer, and *user-generated APPs* (e.g. online games), which are generated by users, based on the API defined by the manufacturer. ICT software enables the realisation of smart functions and acts as the key value creation in the SCP. Its openness determines the success of final product. For example, an open-sourced Android operating system allows users to develop customized APPs.

Scalable design of SCP

At the scalable design stage, the design specifications are ‘stretched’ or ‘shrunk’ based on a set of pre-defined rules (constraints) to satisfy individual user needs (Simpson 2004), which are fulfilled through parametric design optimisation. The optimisation process is undertaken in the physical domain within the fixed product architecture defined by the modular design, such as specific features, specific values of parameter, and trade-offs between performance/solution parameters (Levandowski et al. 2015).

Physical product contains a set of *physical component parameters* corresponding to *configurable physical modules* and *user-generated physical modules*, following the *physical component constraints* pre-defined or approved by the manufacturer, respectively. For example, the diameter of wheel, as a CC, can be optimised within the pre-defined design solution space.

Embedded ICT hardware contains a set of *ICT hardware component parameters* corresponding to the *add-on ICT hardware modules*. The parameters should follow the *ICT hardware component constraints* pre-defined by the manufacturer. For instance, the size of sensors should not exceed 2 cm × 2 cm for the add-on module in a PCB board.

Embedded ICT software contains a set of *ICT software component parameters* corresponding to the *user-generated software APPs*, following the *operating system constraints*. Users can generate various APPs based on the specific programming language (e.g. JAVA for Android APPs), libraries (e.g. jar in JAVA), versions (e.g. Java JDK version), etcetera.

Cyber-physical SCP model

A cyber-physical SCP model serves as the key to enable the real-time information communication between physical and virtual product. A virtual product CAD model with a co-development software toolkit, as *cyber model*, captures the user-generated data for product design innovation. Meanwhile, a physical product with an embedded open toolkit, as *physical model*, provides the tangible experience for users to develop their personalised products. Both the *cyber model* and *physical model* follow the proposed two-stage design process and interact with each other simultaneously through the GUIs as a twin, which enables the real-time elicitation of user-generated data (i.e., both physical design data and data of value), and conversely, the real-time control of the SCP in the real world, as shown in Fig. 7.

Cyber model—virtual product CAD model with co-development software toolkits

The major elements of a cyber-model in a cloud-based environment (Fig. 7) consist of: CAD models of both the SCP

physical and ICT hardware components, and co-development software toolkit, including both the product configuration toolkit, i.e. a knowledge-based system for physical product and ICT hardware modelling, and cloud-based smart function applications developed, i.e. software-as-a-service (SaaS).

At the modular design stage, various changeable CAD part models and SaaS modules should be created through specific APIs and stored coherently online with the modules in the physical model via the Internet. The SaaS modules, especially, as an extension of the embedded smart functions, should be modelled carefully to reflect the physical model, as well as to develop novel applications. For example, a real-time temperature monitoring function in the physical model’s APP is also enabled in the SaaS of the cyber model. Meanwhile, the SaaS can undertake big data analytics from various sources of data to predict weather conditions.

At the scalable design stage, a rule base should be established, including parametric design constraints, smart function application algorithms, and etcetera. For example, a change of shaft length can be detected by the displacement sensor and hence reflected in the cyber-model, to check whether the user is satisfied or not. The evaluated information will be transmitted to the physical model for real-time control.

Physical model—physical product with embedded open toolkits

The major elements of the physical model include: physical product, embedded hardware toolkits with adaptable interfaces, and the embedded software toolkit with APIs in the real world (Fig. 7).

At the modular design stage, each of the configurable or user-generated modules should be equipped with a smart component (e.g. QR code, RFID tag or sensor) for information identification in the cyber model. Also, with the embedded hardware and software modules, a proper method for detecting a change is essential (e.g. utilizing a microprocessor to check the change of signal). This enables coherency in the cyber-physical model. For example, in industrial cases, a workpiece transported to a worker should be scanned by barcode or RFID tag to confirm the information, before the next working step is undertaken.

Correspondingly, at the scalable design stage, due to the characteristics of SCP, only the parametric design of the physical components should be considered, enabled by a set of design constraints defined in either the physical model (e.g. tire size is smaller than the wheel hub) or the cyber model (e.g. the maximum speed should be less than 100 km/h).

Based on the above theoretical model and its descriptions, a case study implementing the proposed approach is illustrated in “[Case study](#)” section.

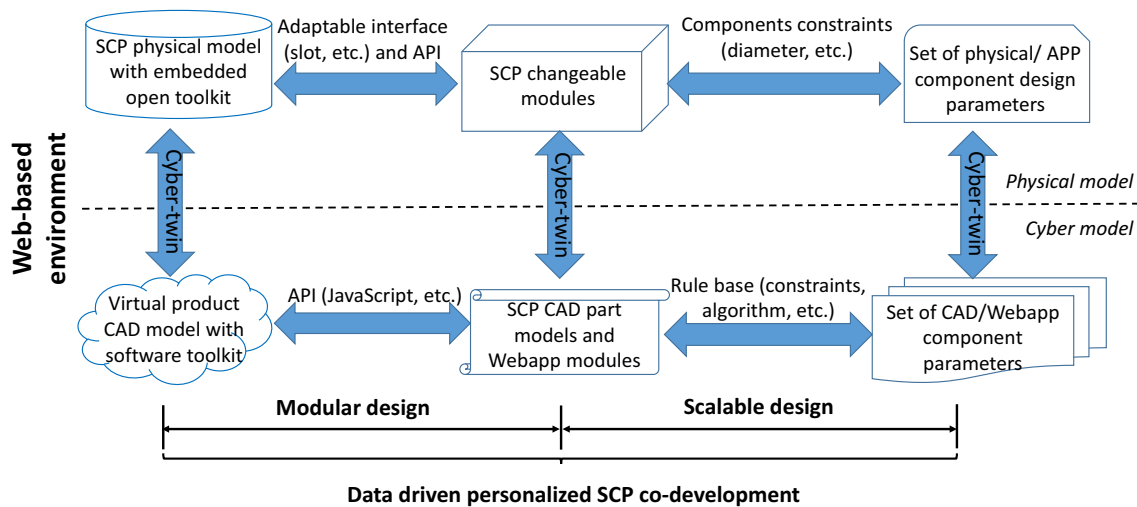


Fig. 7 A proposed data-driven cyber-physical model in a cloud-based environment

Case study

With the rapid development of wireless body sensor network, wearable devices have been increasingly introduced in recent years (e.g. wristbands, shoes, earphones, etc. However, the smart respiratory mask lacks much attention and consequently left almost blank in the SCP market. Traditionally, a respiratory mask was regarded as a mass production product with little change. Due to ever-increasing concerns about air quality and demands for personalisation, several new types of mask have been promoted in the market. Among them, a start-up company in New Zealand is working on developing personalised smart respiratory masks for different application scenarios (e.g. running/cycling masks, anti-smoke masks, and etc.). The data-driven co-development process of a personalized smart respiratory mask: i-BRE, as a typical case of SCP development (Zheng et al. 2017c), is developed for anti-air pollution purposes and presented in this research as a demonstration.

A prototype system has been deployed in the Linux AMI environment hosted by AWS EC2 (2006), which runs on the Apache server and receives HTTP requests from cloud users. An Android smart phone is utilised for the APP development and acts as the bridge for communication between the SCP and the online system. For illustration simplicity, only one case of i-BRE physical/virtual module change (i.e. user-generated data for product development), and one of its APP smart function design (i.e. user-generated data as part of product value), are presented below.

Case one: i-BRE mask physical product prototyping

i-BRE, as a SCP with embedded open toolkits, follows the modular design principles described in “SCOAP modelling”

section, as shown in Fig. 8. The nasal cushion, as the changeable physical module, is chosen as an example to describe the SCP data-driven cyber-physical model in a cloud-based environment.

The system architecture and information flow of i-BRE cyber-physical model is given in Fig. 9. It contains two parts: customised product prototyping by an online PCS (blue arrows) and physical product change by embedded open toolkits (red arrows). For the prior one, user can select different product attributes in the cyber model within the product family scope. The PCS with GUIs is developed based on Wordpress, a free and open-source content management system based on PHP and MySQL, and hosted in an AWS EC2 instance. A set of design constraints are stored in the MySQL rule base, and the virtual CAD models of the physical product is created in an online CAD environment, i.e. Onshape. Updates of CAD data are ensured by the Onshape FeatureScript, a language for parametric design modelling. The two models are connected coherently by transferring XML files to the website. Once user confirmed his/her design, the generated design information stored in the database will be exported to a STL file and transmitted to a connected 3D printer for product prototyping. This information flow enables the rapid prototyping of the personalized SCP.

Reversely, to capture the user generated physical model in the cyber model, in this case study, each physical module is identified by a specific QR code generated by an online QR code generator tool (<http://www.qr-code-generator.com/>). For simplicity, only predefined product attributes are considered and the only information stored in the QR code is the URL of a cyber module (e.g. a nasal cushion CAD model), which retrieves specific product information of the cyber model. An Android device with a QR code reader APP and camera is utilised for scanning the QR code and triggering

Fig. 8 Modular design of i-BRE wearables

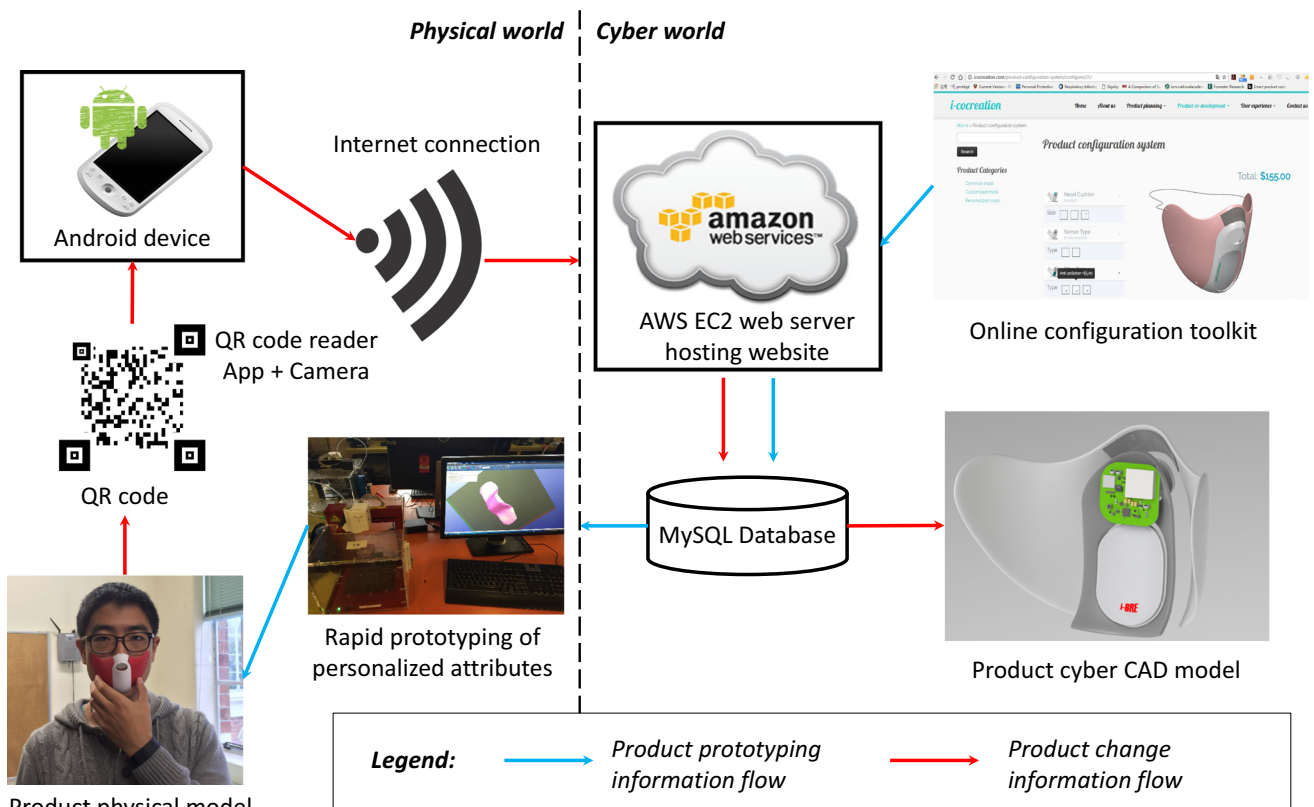
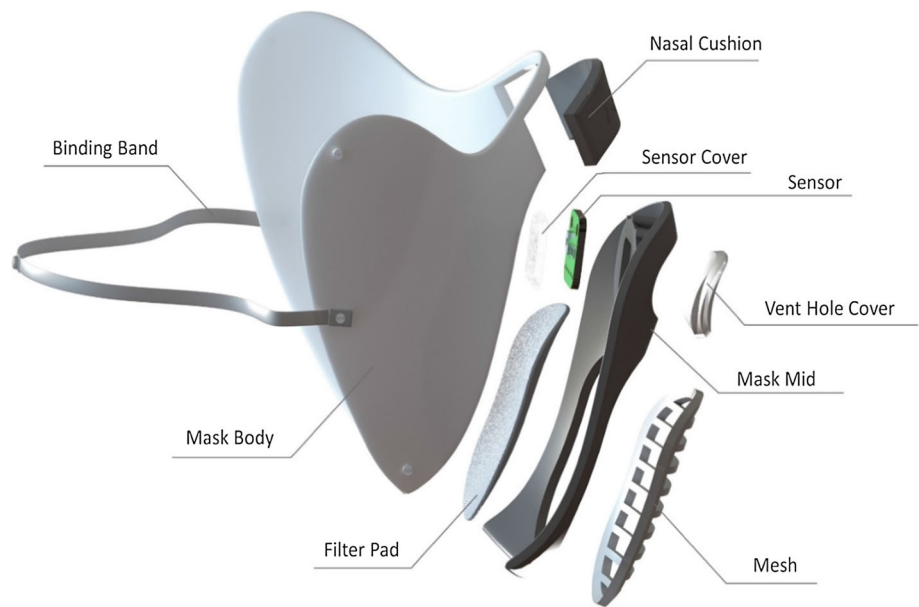


Fig. 9 System architecture and information flow of i-BRE wearable data-driven cyber-physical model

this event by communicating via the Internet. In this manner, once the user changes the nasal module to a bigger size or a different colour in the physical model, a HTTP request will be received by the webserver updating the design information in the cyber model in real-time accordingly. This information flow enables the product change coherently.

Case two: i-BRE mask smart function application innovation

As an anti-pollution mask, one of the key smart functions of i-BRE is to monitor the breathing condition of the user in order to provide a real-time warning to the user about

whether it is worn correctly. To achieve that, this example illustrates how to enable data-driven smart function application co-development in a cloud-based environment.

The system architecture and information flow of i-BRE smart function application is shown in Fig. 10, which contains two parts: breathing condition monitoring (blue arrows) and control (red arrows). For simplicity and robustness, the topology of the example follows a standard Client–Server architecture, in which the client utilises MetaWear sensors (<https://mbientlab.com/sensors/>). One of them, i.e. MetaEnv as the embedded hardware, and user-generated data is transmitted via Bluetooth 4.0 low energy consumption protocol. It is a tiny, yet powerful platform powered by sensor technology (e.g. temperature, pressure), Bluetooth 4.0, ARM and a simple SDK for wearable and connected device applications. Since it is incapable of communicating with the Internet on its own, a Bluetooth 4.0 Low Energy-capable Android device was used as a bridge. The server side was developed in Ruby on Rails with a SQLite3 Backend handled by Rails' Active Record and hosted in a web instance.

One simple way to determine whether i-BRE is correctly worn or not is by monitoring the real-time user-generated pressure data. For the accuracy of data analysis, a sample size of 39 people was selected for this project, and each asked to wear an i-BRE wearable for 2 min in three modes: running, walking and climbing up the stairs. The Ruby on Rails server application handles user registration and access control for the Android app via ActiveRecord and the bCrypt Gem, loaded into Rails. Upon login, the server generates a session and allows users to remain access to the server (Fig. 11). Data storage is handled by Rails' ActiveRecord and its SQLite3 Backend, with the individual breath data being stored in the database with the following fields: *token*, *timestamp*, *pressure* and *user ID*.

The server is able to differentiate between different users by the generation of a unique token for each user, allowing data from each user to be identified. The pressure component of the MetaEnv sensor allows for the following sample rates: 0.99, 1.96, 3.82, 7.33, 13.51, 31.75, 46.51 and 83.33 Hz. It is assumed that a faster, more accurate detection of the breathing cycle improves the user interaction experience. Due to IP issue, elaborate work of sampling process is omitted here. As a result, a sample rate of 46.51 Hz with a Sample Size of 512 is chosen and implemented in the Android device as a smart APP for real-time wear fitness warning (Fig. 12).

Discussions

From the above description with the illustrative case study, one can find that the proposed design approach still follows some basic principles of the classical design theories, namely, (1) the two-stage based product design process or set-based

design principles with platform for lean product development by narrowing down the customer requirement changes step-by-step into a later design stage; (2) the adaptable design principles, which considers co-creation product changeability with a lifecycle concerns; (3) the built-in-flexibility or embedded open toolkit, by postponing the flexibility of product differentiations into the customer domain after manufacturing rather than in the design stage. Nevertheless, as a general method, there exists several novelties beyond the existing design methodologies, enabled by the IT innovation:

1. Different from the traditional platform-based approach, which only takes online configuration/product family design, the proposed one is enabled in a cloud-based cyber-physical model, i.e. interconnections between the cyber and physical spaces. Hence, the design process can be reflected in the cyber space in a remote manner with real-time access to the physical product. Therefore, interactions between users and designers/manufacturers are fundamentally changed as a mapping between user solution space and designer's (illustrated in “[Integration of co-development toolkits](#)” section). Meanwhile, due to the scalable implementation and utilization of the cloud, designers/manufacturers can easily deploy their platform and interact with users/suppliers in a cost-effective manner.
2. Owing to the embedded IT, unlike the conventional design approach which the output largely determined based on designer/expert's subjective evaluation, the SCP design process is conducted in a data-driven manner. The massive user/product generated data serves as the key to for product and service innovation and hence enable the final design success.
3. Moreover, with the interconnections between virtual space and physical space, not only active user expression (e.g. online comments) but also implicit needs can be obtained and analysed in a real time manner in the context of use. Hence, the values generated by users can be personalized as well. For example, the pressure data was utilized to capture user's breathing condition, so as to optimize the design parameters, and provide smart warning services.

This research only looks at the type of product for mass customization and personalization purposes and adopts the smart wearable as the case to reveal its feasibility. Nevertheless, as a generic approach, it can be adopted/adapted to many potential industrial applications. For example, the design approach has been applied in the smart personalized bicycle design in our research group (<http://www.mech.auckland.ac.nz/en/about/ourresearch/research-facilities/LISMS.html>). Meanwhile, many other projects, e.g. personalized smart helmet, medical device,

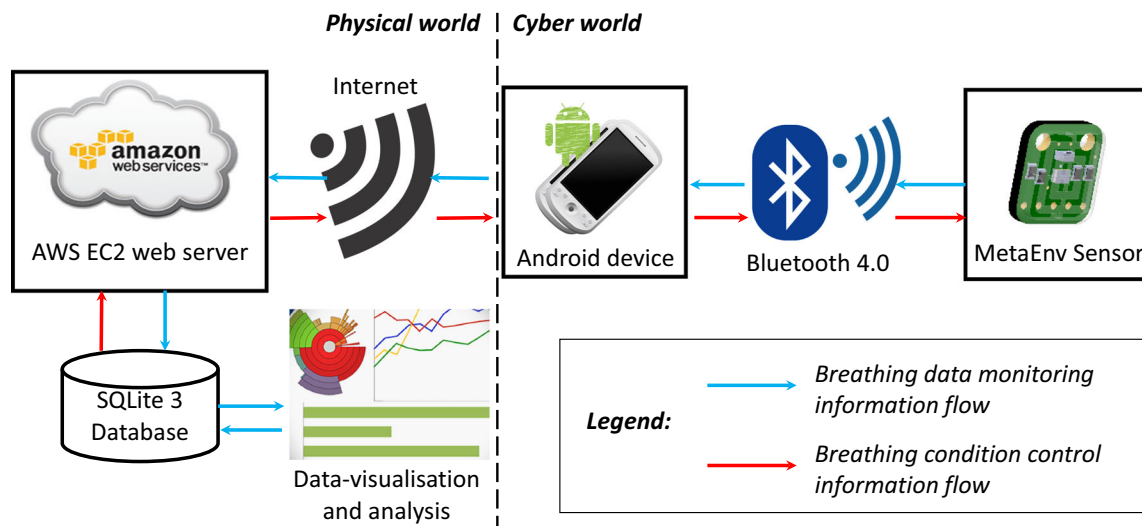


Fig. 10 System architecture and information flow of i-BRE wearable smart function application

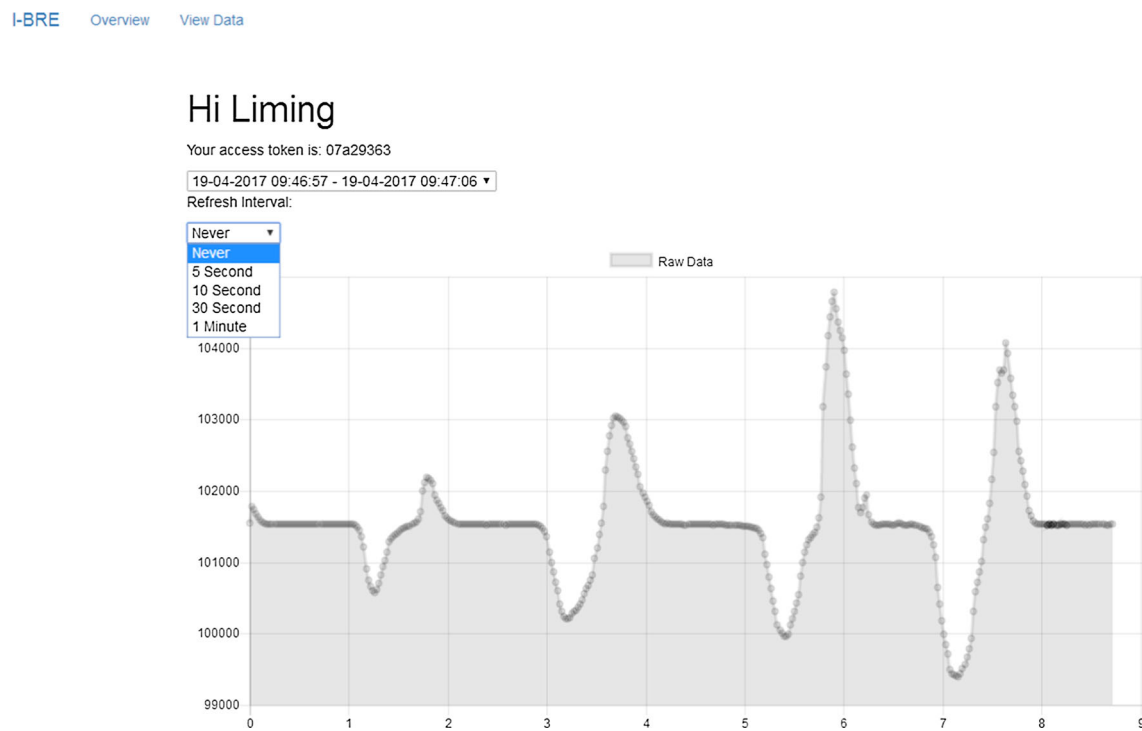


Fig. 11 Screenshot of real-time user data acquisition in a cloud-based environment

smart scooter, to name a few, are ongoing as well. Frankly, one significant challenge lying behind is that IT-driven innovation is still in its infancy, and there are only few existing software components available to support such design approach currently. For example, when PTC publish their IoT-enabled bicycle by Creo 3.0 in 2014, they only had the mock-up without technical support to end users (i.e. software bundle or APIs). Eventually, they introduced Creo 4.0 (now Creo 5.0) recently aiming for interconnecting the physical and cyber space. For our cases, both are developed based

on the development board (e.g. Intel Curie, Metawear) by ourselves and interconnected with the Solidworks/Onshape software through its APIs for data transmission and communications as well.

Conclusions and future work

User generated design data is a prominent data source for data-driven design innovation. Unlike the traditional meth-

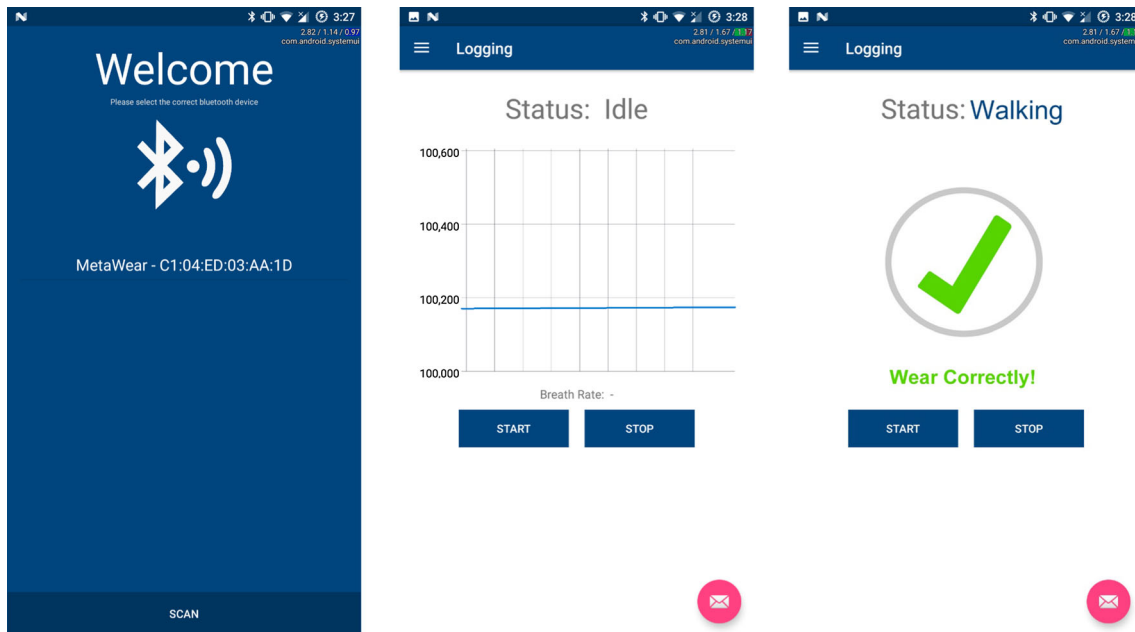


Fig. 12 APP user interface for real-time breathing condition monitoring and control

ods of eliciting design information through users' feedback or marketing analysis, SCP provides a better way to interact with users in a co-development process by capturing the design information in real time. Most existing works focus on the new functionalities or values that are derived from the SCP. This work however intends to establish a systematic way to effectively elicit user generated design data for product innovation by integrating co-development toolkits. Aiming at this, a novel concept of SCOAP is introduced, and a data-driven cyber-physical approach for personalised SCP co-development in a cloud-based environment is further proposed. A case study of a smart respiratory mask co-development process is given with two examples to validate the proposed model. The main contribution of this work lies in two aspects:

1. Introduced a promising systematic design approach with valid case study for early and active user involvement in the co-design/co-creation process with real-time information obtained for next-generation design or service innovation.
2. Proposed a novel concept of SCOAP for SCP open innovation. Its definition, product modelling techniques and cyber-physical product model are given in this work, which can be regarded as a basis for data-driven design for mass personalization in the era of third wave of IT innovation.

Moreover, a brief comparison with other design approaches has been presented, with other potential appli-

cations discussed as well. Due to the page limit, this work only looks at the product development aspect of establishing a generic approach. Many other aspects, such as business model, intellectual property issues, marketing strategies, and so on, are not considered. Also, frankly, this work still has some limitations. For example, this work only considers one product family per user, and the user–user and manufacturer–manufacturer interaction process is not considered, which also plays an important role in a SCOAP environment. Hence, the authors would like to suggest two potential studies in the near future: (1) how to integrate user generated quantitative design data with real-time qualitative user experience data for personalised product innovation; (2) how to analyse various massive users' generated data for SCP family design and platform development in a product ecosystem.

Acknowledgements Authors wish to acknowledge the support from the University of Auckland Human Participants Ethics Committee for conducting the case study (Project No. 018044), and also the funding support from Delta Electronics Inc and the National Research Foundation (NRF) Singapore under the Corporate Laboratory @ University Scheme (Ref. RCA-16/434; SCO-RP1) at Nanyang Technological University, Singapore. The authors are also grateful for the contributions of Mr. Tzu-Jui Lin, Mr. Shiqiang Yu from the Laboratory for Industry 4.0 Smart Manufacturing System (LISMS), the University of Auckland.

References

- Amazon Web Service. <http://aws.amazon.com/>.
 Bénade, M., Brun, J., Brown, I., Le Masson, P., Weil, B., & Piller, F. (2016). How smart products with built in flexibility empower users

- to self-design the use: A theoretical framework for use generation. In *14th open and user innovation conference*.
- Brown, I. (2013). Entre firme et usagers: Des biens génératifs d'usages. *Théorie des biens comme espaces de conception. École nationale supérieure des mines de Paris*.
- Carulli, M., Bordegoni, M., & Cugini, U. (2013). An approach for capturing the voice of the customer based on virtual prototyping. *Journal of Intelligent Manufacturing*, 24(5), 887–903.
- Chen, C., & Wang, L. (2008). Product platform design through clustering analysis and information theoretical approach. *International Journal of Production Research*, 46(15), 4259–4284.
- Chien, C.-F., Kerh, R., Lin, K.-Y., & Yu, A. P.-I. (2016). Data-driven innovation to capture user-experience product design: An empirical study for notebook visual aesthetics design. *Computers & Industrial Engineering*, 99, 162–173.
- Du, G., Jiao, R. J., & Chen, M. (2014). Joint optimization of product family configuration and scaling design by Stackelberg game. *European Journal of Operational Research*, 232(2), 330–341.
- Franke, N., & Piller, F. (2002). Configuration toolkits for mass customization. *Setting a Research Agenda. Arbeitsberichte des Lehrstuhls für Allgemeine und Industrielle Betriebswirtschaftslehre, Technische Universität München*, 33, 4.
- Franke, N., & Piller, F. (2004). Value creation by toolkits for user innovation and design: The case of the watch market. *Journal of Product Innovation Management*, 21(6), 401–415.
- Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design. *Ai Magazine*, 11(4), 26.
- Gross, U., & Antons, D. (2009). Embedded open toolkits for user innovation: Postponing new product development decisions into the customer domain. In *Wirtschaftsinformatik*, 1, 835–840.
- Gu, P., Hashemian, M., & Nee, A. Y. C. (2004). Adaptable design. *CIRP Annals-Manufacturing Technology*, 53(2), 539–557.
- Gu, P., Xue, D., & Nee, A. Y. C. (2009). Adaptable design: Concepts, methods, and applications. *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, 223(11), 1367–1387.
- Gupta, R. K., Belkadi, F., Buegy, C., Bitte, F., Da Cunha, C., Buegin, J., et al. (2018). Gathering, evaluating and managing customers' feedback during aircraft production. *Computers & Industrial Engineering*, 115, 559–572.
- Henkel, J., & Von Hippel, E. (2004). Welfare implications of user innovation. *The Journal of Technology Transfer*, 30(1–2), 73–87.
- Jannach, D., & Zanker, M. (2013). Modeling and solving distributed configuration problems: A CSP-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 25(3), 603–618.
- Kizilcec, R. F., & Schneider, E. (2015). Motivation as a lens to understand online learners: Toward data-driven design with the OLEI scale. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(2), 6.
- Koren, Y., Hu, S. J., Gu, P., & Shpitalni, M. (2013). Open-architecture products. *Cirp Annals-Manufacturing Technology*, 62(2), 719–729.
- Kusiak, A. (2009). Innovation: A data-driven approach. *International Journal of Production Economics*, 122(1), 440–448.
- Lee, E. A. (2008). Cyber physical systems: Design challenges. In *Isorc 2008: 11th IEEE symposium on object/component/service-oriented real-time distributed computing—Proceedings* (pp. 363–369).
- Levandowski, C. E., Jiao, J. R., & Johannesson, H. (2015). A two-stage model of adaptable product platform for engineering-to-order configuration design. *Journal of Engineering Design*, 26(7–9), 220–235.
- Lim, C. H., Kim, M. J., Heo, J. Y., & Kim, K. J. (2018). Design of informatics-based services in manufacturing industries: case studies using large vehicle-related databases. *Journal of Intelligent Manufacturing*, 29, 497–508.
- Lin, K.-Y., Chien, C.-F., & Kerh, R. (2016). UNISON framework of data-driven innovation for extracting user experience of product design of wearable devices. *Computers & Industrial Engineering*, 99, 487–502.
- Ma, J., & Kim, H. M. (2016). Product family architecture design with predictive, data-driven product family design method. *Research in Engineering Design*, 27(1), 5–21.
- Metawear Sensor. <https://mbientlab.com/sensors/>.
- Mosallam, A., Medjaher, K., & Zerhouni, N. (2016). Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27(5), 1037–1048.
- Onshape. <https://www.onshape.com/>.
- Piller, F., Ihl, C., & Steiner, F. (2010). Embedded toolkits for user co-design: A technology acceptance study of product adaptability in the usage stage. In *2010 43rd Hawaii international conference on System Sciences (HICSS)* (pp. 1–10). IEEE.
- Piller, F. T., & Walcher, D. (2006). Toolkits for idea competitions: A novel method to integrate users in new product development. *R&D Management*, 36(3), 307–318.
- Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92(11), 64–88.
- Porter, M. E., & Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, 93(10), 96–114.
- QR code generator. <http://www.qr-code-generator.com/>.
- Rijsdijk, S. A., & Hultink, E. J. (2009). How today's consumers perceive tomorrow's smart products. *Journal of Product Innovation Management*, 26(1), 24–42.
- Salvador, F., & Forza, C. (2004). Configuring products to address the customization-responsiveness squeeze: A survey of management issues and opportunities. *International Journal of Production Economics*, 91(3), 273–291.
- Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(1), 3–20.
- Suh, N. P. (1990). *The principles of design* (Vol. 990). New York: Oxford University Press.
- Trentin, A., Perin, E., & Forza, C. (2012). Product configurator impact on product quality. *International Journal of Production Economics*, 135(2), 850–859.
- Tseng, M. M., Jiao, R. J., & Wang, C. (2010). Design for mass personalization. *CIRP Annals-Manufacturing Technology*, 59(1), 175–178.
- Tuarob, S., & Tucker, C. S. (2015). Quantifying product favorability and extracting notable product features using large scale social media data. *Journal of Computing and Information Science in Engineering*, 15(3), 031003.
- Tucker, C. S., & Kim, H. M. (2009). Data-driven decision tree classification for product portfolio design optimization. *Journal of Computing and Information Science in Engineering*, 9(4), 041004.
- Wang, D. (2011). Robust data-driven modeling approach for real-time final product quality prediction in batch process operation. *IEEE Transactions on Industrial Informatics*, 7(2), 371–377.
- Wang, M., & Chen, W. (2015). A data-driven network analysis approach to predicting customer choice sets for choice modeling in engineering design. *Journal of Mechanical Design*, 137(7), 071410.
- Wang, Y., & Tseng, M. (2014). Attribute selection for product configurator design based on Gini index. *International Journal of Production Research*, 52(20), 6136–6145.
- Wang, L., Youn, B., Azarm, S., & Kannan, P. (2011). Customer-driven product design selection using web based user-generated content. In *ASME 2011 international design engineering technical confer-*

- ences and computers and information in engineering conference (pp. 405–419). American Society of Mechanical Engineers. Wordpress. <https://wordpress.org>.
- Xie, H., Henderson, P., & Kernahan, M. (2005). Modelling and solving engineering product configuration problems by constraint satisfaction. *International Journal of Production Research*, 43(20), 4455–4469.
- Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing*, 28(1), 75–86.
- Yin, S., Wang, G., & Karimi, H. R. (2014). Data-driven design of robust fault detection system for wind turbines. *Mechatronics*, 24(4), 298–306.
- Yu, C., & Zhu, L. (2016). Product design pattern based on big data-driven scenario. *Advances in Mechanical Engineering*, 8(7), 1687814016656805.
- Zhang, L. L. (2014). Product configuration: A review of the state-of-the-art and future research. *International Journal of Production Research*, 52(21), 6381–6398.
- Zheng, P., Lu, Y., Xu, X., & Xie, S. Q. (2017a). A system framework for OKP product planning in a cloud-based design environment. *Robotics and Computer-Integrated Manufacturing*, 45, 73–85.
- Zheng, P., Xu, X., Yu, S., & Liu, C. (2017b). Personalized product configuration framework in an adaptable open architecture product platform. *Journal of Manufacturing Systems*, 43, 422–435.
- Zheng, P., Yu, S., Wang, Y., Zhong, R. Y., & Xu, X. (2017c). User-experience based product development for mass personalization: A case study. *Procedia CIRP*, 63, 2–7.
- Zhou, F., Ji, Y., & Jiao, R. J. (2013). Affective and cognitive design for mass personalization: Status and prospect. *Journal of Intelligent Manufacturing*, 24(5), 1047–1069.