Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

How Well Do Service Concepts Apply to Digital Services and Service Digitalization?

Steven Alter
University of San Francisco
alter@usfca.edu

Abstract

This paper explores the extent to which typical service concepts apply to digital service (DS) and service digitalization. It defines service, service systems, digital, digitalization, digital objects, digital agents, digital service, and service digitalization. Application of those definitions to four real world cases explores how well concepts from the service literature describe DS and service digitalization.

1. Introduction

The title of the HICSS-53 (2020) track on Digital Services and the Digitalization of Service raises many questions. The word *digital* implies that DSs are delivered by digital technologies (DTs), not just by people who happen to use computers. If so, what should stakeholders be able to see or understand about those services beyond user interactions that are not totally digital? Service digitalization sounds like trends or specific changes leading toward performing services through DTs. How visible should those change processes be in organizations and society, especially where service digitalization automates work that may have defined employee careers? Is there any reason to believe that DSs are better than nondigital services or that trends toward service digitalization are beneficial for people and society?

Understanding DSs and service digitalization calls for defining those terms, providing a series of examples (not just isolated, cherry-picked examples that may not be representative), showing whether and how typical service concepts apply to those examples, and reflecting on what those results say about concepts and generalizations about service.

The path toward understanding DSs and service digitalization is complicated by divergent definitions of service and service system by authors describing different types of situations at macro, systemic, and micro levels of detail, e.g., from business models to operational service systems to services that communicate between devices. In addition, terms describing service-related phenomena, trends, and possibilities often seem more metaphoric and utopian than operational or practical.

This paper emphasizes the immaterial nature of DSs that process digital objects (DOs) and the more tangible sociotechnical system changes that rely on DTs. It is based on four assumptions that are consistent with parts but not all of the literature:

(1) Clear definitions and real examples are needed. Definitions should not rely on characteristics that apply to some situations but not others, as when [1, p. 324] noted that characteristics "identified as distinguishing services from goods" (intangibility, inseparability, heterogeneity, perishability) are not satisfactory for defining service. Related questions ask whether characteristics associated with service in general apply to DSs and whether any specific characteristics distinguish DSs from other services.

The existing service discourse relies heavily on ideas such as value proposition and resource integration that often are taken for granted [2, 3] despite seeming distant from operational realities in general and very far from realities of DSs in which automated services interact with each another. Also, an optimistic bias in the most widely cited articles that mention service science (see [4 p. 2]) sometimes is reflected in definitions that include aspirations not achieved in many real world situations, e.g., win-win, creating mutual value, balancing risk-taking and value cocreation, achieving and maintaining sustainable competitive advantage, and satisfying all relevant participants over time. Aspirational goals do not belong in basic definitions.

(2) Ideas about DSs and service digitalization are relevant to all services, not just services for external customers. Concepts such as economic exchange, value proposition, and competition that appear in FP1, FP2, FP4, FP5, and FP7 of S-D logic [5] are more relevant to services for external customers and less relevant to internal IT services and other services for internal customers. Definitions of DSs and service digitalization should recognize that requirements for efficiency and consistency often

URI: https://hdl.handle.net/10125/63886
978-0-9981331-3-3

dictate mandatory, not voluntary, use of internally directed services. Consistent treatment of services for internal and external customers implies that services for the same direct beneficiaries should not flip to non-services after a provider organization merges with a customer organization that no longer pays or exchanges anything for those services.

(3) DSs and service digitalization are shifting the balance between work performed by people with the help of machines and work performed autonomously by machines. Developments related to the changing nature of work (e.g., [6, 7]) tend toward digitalization that supports partial or total automation of both production activities and the use or operation of product/services. Human activities often are replaced or enhanced through activities performed by automated agents operating on behalf of people or organizations either as providers or customers. That leads to questioning tacit or explicit assumptions that service concepts and generalizations are fundamentally about sociotechnical systems.

4) DSs and service digitalization can be beneficial or detrimental to specific stakeholders, either intentionally or accidentally. Widely publicized negative examples such as those in [4, p. 1] demonstrate that positive connotations around the term service should not mask the possibility that DSs and service digitalization may cause harm, either intentionally (as in replacing a competitor) or unintentionally (as in devaluing established skills or through accidents or coincidences). Thus, cheerleading about potential benefits of services should be avoided, or at minimum should be balanced with recognition of risks and downsides.

Goal and organization. This paper explores the related concepts of DS (often a localized type of activity) and service digitalization (a change process often at the level of enterprises or industries).

Goal: Define the terms digital service and service digitalization and use sufficiently different real-world examples of service systems to illuminate whether common service concepts and generalizations describe realities of those situations.

The next section provides definitions in five areas: service, service systems, digital and digitalization, digital objects and digital agents, and DSs and service digitalization. After those discussions, the concepts of DS and service digitalization will be applied in summaries of four real-world cases that include a mix of enterprise systems, platforms, ecosystems, and varying degrees of automation. A final section explores how well selected concepts from the service literature describe those situations.

2. Basic concepts

The terms digital and service both have been used in vastly different ways in different disciplines. Even the meaning of digital should not be taken for granted. Simple Google searches led to a list of 141 two-word phrases (not shown here) that apply different connotations to digital, e.g., digital artifact, digital business, digital culture, digital disruption, digital economy, etc. through digital zombie. The following sections identify and define terms that are needed to understand DSs and service digitalization.

2.1 Service

Service has been defined in many ways. [8] explains how most definitions of service emphasize one of the following portrayals of service: as acts for the benefit of others, as a sector of the economy, as outcomes, as a response to a request, as coproduction, as value cocreation, as economic exchange, and as encapsulated functionalities.

A simple dictionary-like definition, "an act or group of related acts performed to produce or facilitate benefits for others," is natural in everyday business situations such as providing food services, gardening services, or police services. A related definition in S-D logic is "application of skills and knowledge (operant resources) for the benefit of another party." [9, p.6]. Both definitions imply that all economic activities are services, including the production of goods, in turn implying that distinctions between products and services often are not useful for understanding operational services. The definition of service as outcomes applies most directly to controlled, contract-driven situations, such as IT services performed under service level agreements or government services that distribute information to citizens. It applies less well to many human service situations whose outcome depends on joint efforts of customers and providers, such as welfare services, education, or medical care, The encapsulated functionalities definition is appropriate for delegated production of precisely defined outcomes by human or automated agents that will produce those outcomes independently, with no oversight or visibility for the requesting entity. Examples of that type of definition appeared in *IBM Systems Journal* [10,11] and form the basis of USDL, the Unified Service Description Language [12]. Section 2.5 will reflect those points in defining DS.

2.2 Service system

Sources such as [13, p. 76] note that "'service system,' is understood in different ways by the various communities. In some cases, it mainly refers to a set of interconnected services, while in other cases it is used to include other entities besides the service itself, i.e., people, artifacts, resources, the external environment. In these cases, a service system is a complex socio-technical system." A frequently cited definition is "configurations of people, technology and other resources interacting via value propositions to create mutual value." [14, 15]. That definition is difficult to apply to examples shown later, where value propositions and mutual value are elusive, especially where value propositions are unstated and where stakeholder interests conflict.

This paper uses a straightforward definition: a service system is a work system (WS), i.e., a system in which human participants and/or machines perform processes and activities using information, technology, and other resources to produce product/services for internal and/or external customers. The nine elements of the work system framework [16] outline a basic understanding of a WS (and hence a service system). *Processes and activities*, *participants, information*, and *technologies* are completely within the WS. *Customers* and *product/services* may be partially inside and partially outside because customers often participate in work systems. Coproduction occurs when customers participate in a provider's work system. Value cocreation occurs when providers participate in a customer's value producing work system (consistent with [17] but not with an assertion in [5] that cocreation is not optional). WSs operate within an *environment* that matters (e.g., national and organizational culture, policies, history, competitive situation, ecosystems, demographics, technological change, etc.). WSs rely on *infrastructures* shared with other WSs and increasingly tied to vendor platforms and digital ecosystems. WSs should support enterprise and departmental *strategies*.

The *and/or* in the definition of WS implies that a WS, can be sociotechnical (with human participants) or totally automated. For example, accountants making decisions and performing other work related to creating financial statements are participants in a sociotechnical WS that also is an IS (i.e., a WS devoted to processing information [16]). In turn, that sociotechnical IS overlaps with a totally automated IS that stores accounting data, generates reports, and automates other related tasks.

2.3 Digital and digitalization

ICT applications of ideas related to digitized information go back at least to Shannon's theory of communication from 1948, which focused on assuring that the correct message arrives when a binary coded message is transmitted from one machine to another. Thirty years later [18] provided an insight that is more directly related to understanding today's DSs: "Not only does IT process abstract resources (i.e. information) but also the technology is itself partly information. Innovation in IT occurs typically through the production of new information resources including abstract machines in the form of software rather than the development of new types of physical machines."

Negroponte's 1995 book *Being Digital* [19] expressed an overarching view by proposing that the physical world consists of atoms while the digital world consists of bits. … "A bit has no color, size, or weight, and it can travel at the speed of light. It is the smallest atomic element of the DNA of information. For practical purposes we consider a bit to be a 1 or a 0." [19] also noted that "the mixing of audio, video, and data is nothing more than commingled bits." Digitalization exploits the ability to express information and programs as bits.

Digitalization, digital innovation, and digital transformation are overlapping terms related to significant change that relies on digitized information and DTs. The IT glossary of the consulting group Gartner defines digitalization as "the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business." In contrast, [20] says that "the term digitalization [was] coined to describe the manifold sociotechnical phenomena and processes of adopting and using [digital] technologies in broader individual, organizational, and societal contexts." It notes three waves of digitalization: replacing paper with computerized information, introducing the Internet as a global communication infrastructure, and now a third wave with converging SMAC (social, mobile, analytics, cloud) and increasing miniaturization and processing power bringing a vision of ubiquitous computing closer to reality. A HICSS 2019 paper [21] expressing an operational view comes close to the current paper's perspective by seeing digitalization at the work system level and identifying four distinct paths of work system digitalization.

2.4. Digital objects and digital agents

Digitalization relies on DTs and DOs. DTs include electronic devices that execute software and the software that is executed on those devices. DTs process DOs.

Digital object. A DO is a bit stream, i.e., a set of 0's and 1's produced by people or their electronic agents to achieve a purpose. Bit streams are immaterial and do not take a physical form even though they may instruct physical devices to produce things that have a physical form. All text, audio, image, and video objects can be expressed in bits. Thus, bit stream is like a common denominator allowing similar technical methods to store and transmit different types of data.

DOs are not information in the sense of informing people because strings of zeros and ones are not designed to inform people. For that purpose, DOs must be translated into a format (e.g., numbers and letters for text) and inscribed onto a medium such as a screen that makes them intelligible. In contrast, DOs such as messages passed between DSs are quite useful without informing people about anything. In effect, they are information to the DSs. In an increasingly automated world, it is increasingly less useful to conflate information with interpretation by people (see views of information in [22,23]).

The fact that DOs are bit streams does not imply that DOs are inherently simple, as demonstrated by examples of DOs: the number *pi* to 1000 decimal places, a message from one machine to another, a digital photograph, a book's content, an invoice in a corporate database, a website, a blog, the content of Google Scholar, software that runs a game, and the Windows 10 operating system. A deeper look at DOs (beyond the current scope) might include a DO framework that covers the content of the DO along with relevant metadata, access rights, syntax, interoperability guidelines for various devices, and history of modification or usage.

Some DO's are inherently static, whereas others such as blogs and websites evolve over time, leaving important questions about exactly which version is being viewed or analyzed. A related theory of DOs [24] focuses on generic properties such as editability, interactivity, openness and distributedness, all of which are linked to modularity and granularity. Other DO attributes in [25] include programmability, accessibility, communicability, accessibility, transfigurability, traceability, and non-rivalry (possibility of simultaneous usage). Those attributes and others often provide valuable affordances.

Digital agents. These are totally automated work systems whose software controls the operation of DTs that capture, transmit, store, delete, manipulate, and/or display DOs. Digital agents perform work on behalf of a user (person or organization) or another digital agent. The DO nature of software in digital agents makes them much more changeable than physical devices (for better and for worse). Small digital agents control the performance of unitary tasks such as performing small calculations or transmitting messages Much larger digital agents are immense systems such as the Google search mechanism or AI-based translation systems. Activities performed by digital agents may be directed toward DOs (e.g., translate a sentence) or may control digital devices, as when a digital agent instructs a device to display a page.

Digital agents may have many limitations. They may or may not perform work as expected (like human agents performing work for someone else). They may operate based on incorrect inputs. Their software may contain bugs, may ignore important factors, may be unable to recognize or respond to exceptions, and may not operate at all under various circumstances. Contrary to common AI hype, today's digital agents have no real understanding of the context or semantics of the DOs that they process.

For current purposes digital agents can be viewed as equivalent to DSs that perform tasks requested by human users or by other digital agents. The notion of digital agent was introduced mainly as a reminder that both human and technological agents may not meet expectations when performing assigned tasks.

2.5 Digital service and service digitalization

A paper [26] summarizing the HICSS 2019 minitrack on Digital Services and Digitalization of Service defined DSs as "systems that enable value co-creation and limit value co-destruction through the development and implementation of information communication technology (ICT) enabled processes that integrate system value propositions with customer value drivers." That definition 1) does not say whether a DS can be a sociotechnical system with human participants, which would allow almost any service system that uses IT to qualify as a DS, 2) includes development and implementation within the definition of DS, and 3) includes the opaque characteristic of "integrating system value propositions with customer value drivers."

A different HICSS 2019 paper [14] defined a DS as "a service executed in full by a technical system, when a user invokes a digital Information, Computing, Communication and Automation Technology (ICCAT) based system that (co-)creates the desired outcome. … [in a DS] the assistance or benefit is mediated by means of ICCAT system between the service provider and service user roles. … producing a result or product is done by means of automated processes based on ICCAT system."

The following is a simpler definition of DS: a totally automated WS whose software-controlled activities produce and/or deliver DOs. DSs often serve as components in a hierarchy of interacting DSs which may play digital agent roles in sociotechnical systems. ICCATs [14] serve as DSs that coordinate other DSs to produce results for human or automated customers. DSs serving as digital agents may be triggered by human requests, by messages (DOs) sent from other digital agents (such as ICCATs), or by clock time or other conditions.

The smallest DSs are imperceptible micro-services that are components of larger DSs designed as interacting micro-services to enhance software clarity, testability, and reusability. DSs that are directly or indirectly perceptible to people may control display devices, perform computations, capture images in digital cameras, provide Internet service, provide real-time feedback for people based on data captured and stored as DOs, and so on. The largest DSs might apply numerous digital agents and devices to perform complex services such as controlling a factory or providing Internet service across a large geographical area.

DSs as work systems/ service systems. The work system framework can be used to describe a DS and its context. A DS's *customers* may be people or digital agents of people. A DS's *product/services* are DOs that may be directed to other DSs or to devices. Its *processes and activities* are controlled by software. DSs have no human *participants*. DSs may be triggered by inputs or requests from people whose inputs or requests may be mediated by DSs for user interaction. *Information* produced and used by a DS consists of DOs that may or may not be perceptible by human users. *Technologies* in a DS are DTs that perform combinations of capturing, transmitting storing, retrieving, deleting, manipulating, and displaying DOs. A DS's *environment* is the human needs and conditions that trigger its execution plus the state and activity of devices, software, and other resources external to the DS that affect its operation. A DS's *infrastructure* includes resources shared with other DSs, often as part of a technical platform or digital ecosystem. A DS's *strategy* is its architecture and other aspects of its design rationale.

Degrees of visibility. DSs have different degrees of visibility to users and other stakeholders. DS activities may be hidden purposefully from users and stakeholders consistent with information hiding, a programming technique for controlling complexity of software and increasing reliability and reusability. A DS may be visible to varying degrees:

Invisible. A DS is an internal component of a larger DS provided by an outside entity. Lack of visibility for programmers may be risky, as when outside entities use components from other entities that may use components of questionable quality.

Visible to programmers but not to users. A DS is an internal component of a larger DS. Programmers may need to understand it in depth, even if it has no direct interest to stakeholders.

Semi-visible to users. A DS expresses parameters or business logic that may matter to stakeholders who are not interested in exactly how those parameters or business rules are built into software.

Highly visible to users. Visibility is necessary because understanding of important details of a DS is essential for execution of stakeholder responsibilities.

Service digitalization. Digitalization of service systems is a change process (and trend) of increasing the degree of reliance on DSs and DOs in service systems, often with the effect of increasing the degree of automation in those systems. Current jargon makes few real distinctions between service digitalization, digitalization in general, digital innovation, digital transformation, and many other ways of describing greater reliance on technology and sometimes greater control of human workers. Four examples discussed next can be used to visualize the role of DSs and service digitalization.

3. Four cases illustrating digital services and service digitalization

The description of each example will mention aspects of the situation that are directly relevant for current purposes related to DS and service digitalization and will omit many details that would provide greater insights about the situations (but not within the context of a 10-page paper that covers other ideas). The four cases cover service systems that range from totally automated to highly intensive in human interactions. They will be identified using pseudonyms: AdEx, MCAS, EMR+, and SMgt.

3.1 AdEx: a totally automated ecosystem for placing ads in online publications

AdEx [27] is a totally automated ecosystem that controls the insertion of ads into web-based content such as online news articles. "It is a huge, real-time bidding process, whereby ads are automatically assigned to media spaces across types of media and geographic regions upon an individual user's browser request. ... the entire ecosystem's exchange with its hundreds of platforms operates 'on-demand' every time a user's browser opens a publisher website and triggers a real-time request for an ad. The whole exchange is usually completed under 100 ms and remains entirely invisible to the user who may experience a small lag in loading the publisher page."[27]

The number of parties involved when a browser requests a webpage makes the complexity and speed of AdEx's response all the more impressive. A user's request to view a page triggers totally automated activities involving "platforms that act on behalf of traditional actors such as online publishers and marketers (buyers, sellers) and novel actors such as various data brokers and intermediaries. Demand Side Platforms (DSP that act on behalf of marketers), Supply Side Platforms (SSP that act on behalf of publishers) and many Data Management Platforms (DMP) exchange massive flows of data in real time as they seek to buy individual user attention."

The service system. This is an ecosystem involving actions and interactions of hundreds of platforms representing different interests. It produces bids that are accepted or rejected. Its ad exchange process considers all bid responses for a given bid request and declares as the winner the highest bid that fits all requested parameters related to location on the page, size, and other parameters. The winning bid is executed by placing an ad into the web page seen by a reader. Although people maintain this system, its operation is totally automatic.

Service digitalization. AdEx is a long-term innovation that occurred over more than a decade as part of a revolution in the advertising industry. Achieving the automated and real-time nature of ad exchanges required establishing standards and technologies to govern the behavior of market participants (DSP, SSP and SMP) and deal successfully all related data processing tasks. The related protocols and standards were developed by a group of demand-side and supply-side platforms.

Digital services. The entire totally automated ecosystem can be viewed as a gigantic DS. In turn, that very large DS consists of many smaller DSs that deal with obtaining offers and bids, selecting the best bid, serving the ad, and reporting the results to stakeholders. Many of those smaller DSs can be subdivided into micro-services that perform small tasks that are meaningful only to programmers.

Digital objects. These include cookies, bid request objects, bid response objects, impression objects (the ads themselves), and so on.

3.2 MCAS: Maneuvering characteristics augmentation system

Boeing 737 Max airplanes include an MCAS that automatically lowers the aircraft's nose in very rare situations when the combination of airspeed, altitude, and angle-of-attack indicates that a stall may be imminent. Inquiries into two fatal crashes of Boeing 737 Max jets in 2019 pointed to problems related to the MCAS. Both crashes occurred soon after takeoff, and pilots seemed to be fighting with the MCAS, trying to increase altitude while the MCAS was lowering the aircraft's nose.

An abbreviated version of the relevant background starts with Boeing's decision to build a new version of its popular 737 aircraft. The new version had larger engines and different aerodynamic properties, despite which Boeing argued that pilots who had flown the 737 would not have to undergo expensive retraining. When the 737 Max seemed not to operate smoothly in high-speed maneuvers on a flight simulator, Boeing addressed the problem by using software fixes to the MCAS instead of the much more expensive approach of changing physical aspects of the plane. Subsequent flight tests found that the Max was not operating well while in near stall conditions at low speed. Boeing decided to incorporate further changes into the MCAS. The US Federal Aviation Agency had approved the previous version of the MCAS and did not examine the new version. Flight tests with the new MCAS in various flight situations seemed successful, but those tests did not consider the possibility that the angle-of-attack data might be wrong. The previous version of the MCAS used two sensors. "In the 737 Max, only one of the flight management computers is active at a time—either the pilot's computer or the copilot's computer. And the active computer takes inputs *only* from the sensors on its own side of the aircraft." [28] Using inputs from only one sensor proved a risky approach because FAA databases included hundreds of examples of bent, cracked, sheared-off, poorly installed or otherwise malfunctioning angle-of-attack sensors on commercial aircraft over three decades [29]. Communication regarding the new MCAS software within different groups in Boeing and between Boeing, its customers, and the FAA seemed incomplete. Many people seemed surprised that the MCAS used only one sensor [28, 29]

The service system. Modern aircraft are complex cyber-physical systems controlled through software. The aircraft, its crew and operations personnel might be viewed as a service system for delivering passengers to destinations. The MCAS is an automated service system that tracks the airplane's flight status and avoids stalls by raising the airplane's nose automatically (even overriding the pilot).

Service digitalization. The MCAS was originally designed as an automated safety system augmenting other safety systems. The revised MCAS seemed at the same level of service digitalization.

Digital services. The MCAS performs the DS of tracking flight status and directing the airplane to bring its nose up in rare near-stall conditions.

Digital objects. These include airspeed, altitude, angle-of-attack, and probably other data not mentioned in the sources consulted.

3.3 EMR+ an electronic medical record system

EMR+ is described by a surgeon author in a case study called "The Update: Why Doctors Hate Their Computers" [30]. This case involves a large scale electronic medical records (EMR) implementation in a medical group with 70,000 employees. The chief clinical officer who supervised the upgrade from previous software saw important benefits in standardization and in benefits for patients, who now could have access to their own medical records and could send emails to their physicians. The author had a different view, saying "I've come to feel that a system that promised to increase my mastery over my work has, instead, increased my work's mastery over me". The case reports that many primary care physicians suffer from burnout because they need to devote so much time to entering data into computers, often doing that from home at night. The case noted "signal fatigue," saying "just ordering medications and lab tests triggers dozens of alerts each day, most irrelevant, and all in need of human reviewing and sorting." A primary care physician described erasing EMR alerts and emails that were overwhelming. Those included automated email reminders that previous emails had not been answered. Contrary to expectations about better communication, the author "began to see the insidious ways that the software changed how people work together. They become more disconnected, less likely to see and help one another, and often less able to [help]." A medical support worker described being denied access rights to information that she formerly organized to help physicians work more efficiently. A surprising adaptation for some physicians was a new work role, a "scribe" who attended patient visits and entered data to offload that burden from the physician.

The service system. This case discussed two service systems that overlap, a service system of providing medical care and a service system of providing and receiving EMR information. The overlap occurs where physicians participate in both systems during patient visits. Attending to the EMR draws physician away from attending to patients (as has been discussed widely) and creates stressful situations when physicians do not have enough time to complete documentation and take care of patients.

Service digitalization. Moving to the new EMR software was challenging and expensive in this large enterprise even though the software existed and had been used in many other medical groups.

Digital services. Providing medical care is not a DS because the work is done by doctors with the help of technology. The EMR software specifies a series of DSs that capture, transmit, store, retrieve, display, and manipulate patient information. Those DSs guide every interaction of a physician with EMR+ and the storage and retrieval of all patient information.

Digital objects. These include patient medical records, physician schedules, and much other information needed to operate the organization.

3.4 SMgt: sales management system using spreadsheets to work around corporate ERP

SMgt is a spreadsheet-based sales management system that is a workaround of a corporate ERP system in a globally distributed textiles company.[31] A corporate initiative implemented a widely used ERP package to achieve greater coordination and control across the entire company. Unfortunately, the flow logic built into the ERP software was unusable in Hong Kong sales branches. The software assumed that stores send inventory orders to headquarters, that headquarters fulfills orders, that receipts go into an inventory area, that walk-in customers buy products, and that stores submit replenishment orders.

That straightforward logic conflicts with physical realities in Hong Kong, where the stores cannot afford inventory areas. Replenishment orders go to a central warehouse shared by four stores that can only show products to customers. Salespeople determine a delivery date from the warehouse that works for the customer and for a delivery service. The global ERP software cannot accommodate that process. A store that sells an item can update its own database, but the warehouse database is not updated until the item leaves for customer delivery, sometimes several days later. Avoiding discrepancies between databases requires separate identification of saleable inventory, items sold but not yet delivered, and returned items. Previously used software addressed that issue and many others that the new ERP system could not handle. When the older software was turned off, the Hong Kong staff created extensive workarounds based on spreadsheets that were invisible to the mandated ERP system. These workarounds allowed them to maintain information about inventory status, items to be delivered, vans that would be required, delivery addresses, and payments to delivery drivers.

The service system. The service system is a sales management system that enables the stores to manage inventory, sales transactions, and deliveries despite significant conflict with the corporate ERP system. It uses spreadsheets to work around those conflicts. Entering and using information helps service system participants coordinate the efforts of sales, logistics, and administrative staff playing different roles.

Service digitalization. This case involves two instances of service digitalization: the corporate innovation of moving to the global ERP and the local innovation of using spreadsheets to execute the sales process efficiently and effectively in Hong Kong. The first innovation was based on widely used ERP software. The second was based on Excel.

Digital services. The ERP system performs DSs such as transmitting replenishment orders and performing receipt and sale transactions. The spreadsheets perform DSs including storing and displaying inventory and delivery status.

Digital objects. These include replenishment orders, warehouse receipts, inventory status for each item, and delivery commitments.

4. How well do service concepts apply to the four examples?

The four examples can be used to test how well service concepts apply to DSs and service digitalization. Each example identified one or more service systems, noted how service digitalization applied, and identified relevant DSs and DOs.

The examples covered a mix of enterprise and non-enterprise systems, platforms, ecosystems, and varying degrees of automation and user interaction. DSs in AdEx and MCAS automated important decisions. DSs in MCAS, EMR+, and SMgt had tangible effects on the work of users, whereas AdEx produced consequential outputs automatically. The EMR+ and SMgt examples involved enterprise-level ISs; EMR+ was used during medical appointments and SMgt workarounds were used to bypass corporate ERP. Three examples involved platforms (AdEx, EMR+, SMgt), Three (MCAS, EMR+, SMgt) involved DSs that failed disastrously (MCAS) or experienced major problems (EMR+ and SMgt).

This section continues by looking at how well service concepts (many of which appear in the 11 foundational premises of S-D logic [5]) apply to DSs and service digitalization in the cases.

Service and service system. The DSs in the cases conformed with this paper's definition of service and service system. Alternatives discussed in [8] posit characteristics that apply to some DSs but not all.

Digitalization and service digitalization. All four examples involved digitalization. There was no hint that a more restricted concept of service digitalization would add meaningful clarity or nuance.

Exchange. The case descriptions do not address S-D logic's treatment of service as exchange. A WS perspective highlights economic exchange only when needed for understanding important opportunities or situation-specific issues, rather than economic or service exchange in general. For example, the EMR+ case mentioned many operational problems and opportunities but said nothing about how patients, insurance companies, or other payers paid for medical services or how doctors were paid.

Goods vs. services. This distinction was not mentioned in the cases. Referring to outputs of service systems as product/services is based on the assumption that distinctions between products and services (or between goods and services) are not useful for understanding operational systems.

Operant resources vs. operand resources. This distinction does not help in understanding DSs that serve as operant resources but often are treated as operand resources based on characteristics of DOs and DSs. The association of knowledge and skills with operant resources does not ring true here since DSs do not exhibit human knowledge and skills even though knowledge was used to create them. Also, for EMR+ the operant versus operand distinction may direct attention away from seeing a patient as an operant resource whose ability to communicate and cooperate matters greatly.

Beneficiaries. All of the service systems have intended beneficiaries. Intentions were not realized when pilots could not overcome MCASs, physicians did not have enough time to enter data (EMR+), and corporate ERP was not usable locally (SMgt). Also, the assumption that "value is always uniquely and phenomenologically determined by the beneficiary" (FP10 in S-D logic) seems distant from situations where DSs are invisible to end customers.

Value cocreation. This idea has been debated by service researchers. [17] says that value cocreation is optional, contrary to the view in the 2016 version of S-D logic, which says "cocreation of value, unlike co-production, is not optional."[5] That extension into the broader realm of institutions and ecosystems says "value cocreation is developing into one of resource-integrating, reciprocal-service providing actors cocreating value through holistic, meaning-laden experiences in nested and overlapping service ecosystems, governed and evaluated through their institutional arrangements" [5, p.7] Those ideas apply most directly to the AdEx ad exchange, which takes the form of an ecosystem. They apply to some aspects of the EMR+ case, where "meaning-laden experiences" are not uniformly pleasant and where the idea of resource integration is opaque. (How many physicians or patients would say that medical care involves integrating resources?) The SMgt case is about workarounds that bypass impractical institutional arrangements. Value cocreation seems unrelated to the MCAS case.

Relational nature of a service-oriented view. It is unclear whether value cocreation should imply some visibility of what cocreators are providing or doing in these cases, especially for DSs that operate automatically and are designed to minimize visibility. More generally, the relational nature of a service-oriented view is not apparent in ecosystems containing actors who may be unaware of each other's existence or contribution to those ecosystems.

Value proposition. A 2017 *JAMS* article on the *customer value proposition* [2] reviewed the history of *value proposition* and produced a preferred enterprise-level definition that has little relevance to DSs: "A customer value proposition (CVP) is a strategic tool facilitating communication of an organization's ability to share resources and offer a superior value package to targeted customers." An earlier definition of value proposition that was cited is more somewhat more appropriate for DSs: "a statement of benefits provided and the total costs for a product." Once again however, this definition does not reflect the nature of many DSs, especially those directed inward.

The value proposition for buyers and sellers in the AdEx data exchange is totally straightforward, i.e., participation is mandatory if they want to buy or sell ad placements. Misleading value propositions at a pre-sales level in the MCAS case tried to convince 737 Max buyers that the Max would not require additional pilot training. The value proposition for the MCAS itself was surely that it would help in an emergency. The EMR+ case involved corporate level value propositions from the software vendor and internal value propositions to get physicians engaged. The case mentioned practical issues that sugar-coated value propositions could not have mentioned. In SMgt a corporate-level value proposition of consistency and control conflicted with a local value proposition of satisfying local customers. FP7 of the 2016 version of S-D logic [5] says "actors cannot deliver value but can participate in the creation and offering of value propositions." The interpretation of the cases through the lens of FP7 is unclear, e.g., most physicians in the EMR+ case likely viewed themselves as providing value rather than just value propositions. The applicability of FP7 to the automated actors in the AdEx case is also unclear.

Resource integration. In combination, the 25 descriptions of resource integration in a review of 57 related articles since 2004 showed that resource integration "usually refers to an empirical phenomenon, without a clear definition or description. Some definitions appear … but there is a definite lack of consensus. Competencies are identified as a prerequisite of resource integration, and resource integration is presented as part of actors' value cocreation efforts and processes. Intuitively, the nature of resource integration may be such that scholars assume the name itself is equivalent to defining it; 'integration' means combining into a whole, so resource integration is self-evidently combining resources into something new. Many publications also define or describe integration tautologically, as the act of integrating, which cannot contribute to theorizing resource integration." [3, p. 4]

Based on the above, it is not clear what resource integration means in relation to the DSs in the four cases. Assume that physicians and patients integrate resources during medical exams or that physicians integrate resources with EMR+. How would they or outside observers know that resource integration occurred? More broadly, the quotation above implies that resource integration means little beyond a universal and largely tacit expectation of using knowledge, skill, and other available resources while cooperating with colleagues, consultants, customers, and suppliers. It is not clear how that tacit expectation provides insights related to DSs.

5. Conclusion

This paper's goal was to define DS and service digitalization and to use diverse real-world examples to illuminate the extent to which common service concepts and generalizations describe realities of those situations. The general conclusions are as follows:

1) The mix of cases seemed adequate for an initial exploration of whether concepts associated with service and service systems apply in valuable ways to DSs and service digitalization. Each of the cases raised issues that were not raised by other cases.

2) Proposed definitions of DS and service digitalization seemed to fit well and benefitted from not being encumbered by characteristics that apply in some cases but not others.

3) Applicability of service concepts to DS and service digitalization can be tested, at least initially, by using previously published examples.

4) Ideas in S-D logic that seem interesting and important when applied to economics and marketing are less useful for understanding DSs and service digitalization.

5) Limitations of this paper start with the fact that it could only apply its ideas to four abbreviated cases. A larger set of more fully described examples likely would reveal at least some issues that the four brief case descriptions do not touch. Future extensions of this research might describe DS-related concepts in a deeper way and might provide detailed comparisons

with related concepts from other DS research and other service research. For example, it could present representative definitions of value cocreation, value proposition, and resource integration and could say more about how well different definitions apply to DSs in accounts of real-world service situations.

6) A final point is that some researchers might be dissatisfied with this paper's definitions and with this paper's mix of cases. One of the most effective ways to move this area of research forward is for other researchers to show how other views of the same or similar ideas lead to richer descriptions of real-world situations related to DS and service digitalization. Those topics are increasingly important and deserve much additional research.

References

[1] S. L. Vargo and R. F. Lusch. "The four service marketing myths: remnants of a goods-based, manufacturing model." Journal of Service Research, 6(4), 2004, pp. 324-335.

[2] A. Payne, P. Frow, and A. Eggert. "The customer value proposition: evolution, development, and application in marketing." Journal of the Academy of Marketing Science 45(4), 2017, pp. 467-489.

[3] R. Findsrud, B. Tronvoll, and B. Edvardsson. "Motivation: The missing driver for theorizing about resource integration." Marketing Theory 18(4) 2018. :pp. 493-519.

[4] S. Alter "Service System Axioms that Accept Positive and Negative Outcomes and Impacts of Service Systems, " Proceedings of ICIS 2017

[5] S. L. Vargo and R. F. Lusch, "Institutions and axioms: an extension and update of service-dominant logic," Journal of the Academy of Marketing Science 44(1), 2016, pp. 5-23

[6] A. McAfee, and E. Brynjolfsson. Machine, platform, crowd: Harnessing our digital future. W. W. Norton, 2017.

[7] A. Richter, et al. "Digital work design,". Business & Information Systems Engineering, *60*(3), 2018, pp.259-264

[8] S. Alter, "Answering key questions for service science,". Proceedings of ECIS 2017.

[9] S. L. Vargo and R.F Lusch, "Service-dominant logic: continuing the evolution." Journal of the Academy of Marketing Science, *36*(1), 2008, pp.1-10.

[10] A.W. Brown, et al. "Realizing service-oriented solutions with the IBM Rational Software Development Platform," IBM Systems Journal, 44(4), 2005. pp. 727-752.

[11] L.G. Cherbakov, et al. "Impact of service orientation at the business level," IBM Systems Journal, 44(4), 2005. pp. 653-668.

[12] D. Oberle, A. Barros, U. Kylau, and S.Heinzl. "A unified description language for human to automated services." Information systems, 38 (1), 2013, pp. 155-181.

[13] R. Ferrario, et al. "Service System Approaches." In Handbook of Service Description, pp. 75-109. Springer, Boston, MA, 2012.

[14] D. Pakkala and J. Spohrer. "Digital Service: Technological Agency in Service Systems." HICSS, 2019.

[15] P.P. Maglio, S.L. Vargo, N. Caswell and J. Spohrer, "The service system is the basic abstraction of service science", Information Science and e-Business Management, 2009, pp. 395 - 406.

[16] S. Alter, "Work System Theory: Overview of Core Concepts, Extensions, and Challenges for the Future," Journal of the Association for Information Systems, (14), 2013, pp. 72-121.

[17] C. Grönroos, "Value co-creation in service logic: A critical analysis." Marketing Theory, 2011, pp. 279-301.

[18] P. Monk, "Characteristics of IT innovation," Journal of Information Technology, 2(4), 1987, pp.164-170.

[19] N. Negroponte, Being Digital. New York: Alfred A. Knopf, 1995.

[20] C. Legner et al., "Digitalization: opportunity and challenge for the business and information systems engineering community. Business & Information Systems Engineering. 59(4), 2017, pp. 301-308.

[21] V. C. Wolf, C. Bartelheimer, and D. Beverungen, "Digitalization of Work Systems—An Organizational Routines' Perspective." HICSS 2019.

[22] E. H. McKinney, and C. J. Yoos, "Information about information: A taxonomy of views." MIS Quarterly, 34(2), 2010, pp. 329-344.

[23] S.K. Boell, "Information: Fundamental positions and their implications for information systems research, education and practice." Information and Organization, 27(1), 2017, pp.1-16.

[24] J. Kallinikos, A. Aaltonen, and A. Marton, "A theory of digital objects." First Monday, 15(6). 2010.

[25] J. Kallinikos, A. Aaltonen, and A. Marton, "The Ambivalent Ontology of Digital Artifacts," MIS Quarterly, 37(2), 2013, pp. 357-370

[26] T. Tuunanen, T. Böhmann, and O. Henfridsson. "Introduction to the Minitrack on Digital Services and the Digitalization of Services." HICSS. 2019.

[27] C. Alaimo, and J. Kallinikos, "Objects, Metrics and Practices: An Inquiry into the Programmatic Advertising Ecosystem." Working Conference on Information Systems and Organizations (pp. 110-123). Springer, Cham, 2018,

[28] G. Travis, " How the Boeing 737 Max Disaster Looks to a Software Developer," IEEE Spectrum, 18 April 2019

[29] J. Nicase, N. Kitroeff, D. Gelles, and J. Glanz, "Boeing Built Deadly Assumptions Into 737 Max, Blind to a Late Design Change, New York Times, June 1, 2019.

[30] A. Gawande, "Why doctors hate their computers." The New Yorker, 2018.

[31] R. M. Davison, L.H. Wong, S. Alter, and C. Ou, "Adopted Globally but Unusable Locally: What Workarounds Reveal about Adoption, Resistance, Compliance, and Noncompliance." Proceedings of ECIS 2019.