iS CHANNEL

The Generativity of Messaging Platforms: A Case Study on Facebook Messenger and Chatbots

Simon Draxinger
MSc in Information Systems and Digital Innovation
Department of Management
London School of Economics and Political Science

KEYWORDS

Generativity
Platform
Modularity
Chatbots
Instant messaging
Facebook Messenger
Machine learning
Natural language processing

ABSTRACT

This paper explains the integration of chatbots into messaging applications as a potential outcome of digital platforms' architectural principles and their characteristic of generativity. A case study of Facebook Messenger (FM) is adopted and analysed through the theoretical framework of layered modular architecture (LMA) proposed by Yoo et al. (2010). To understand the increasing diffusion of chatbots as a consequence of a platform's characteristic of generativity, this essay answers two distinct research questions (RQ) that are as follows: (1) How does layered modular architecture explain generativity on Facebook Messenger?; (2) What are the implications of automation in the form of chatbots on generativity?

1. Introduction

Chatbots offer businesses the chance to increase their customer outreach in unprecedented ways. Chatbots are automated programs which are incorporated into messaging applications. They are based on Artificial Intelligence (AI) to communicate with users, thereby providing a set of services and functionalities. Chatbots are deployed in a range of areas: from simple assistance, such as news coverage, weather forecasts and shopping assistance to business purposes such as customer service (Schlicht, 2016).

Using chatbots, enables a firm to instantly reach a great amount of people since messaging platforms are characterised by large user bases (Schlicht, 2016). Facebook Messenger (FM) for instance is used by over one billion people (Constine, 2016). Furthermore, according to The Q2 2016 Sprout Social Index (2016), 35.5% of respondents prefer social media as the best way for customer service, which constituted the top choice. Additionally, the Index indicates that the average number of social messages awaiting a response grew 18% in comparison to 2015, with brands' responding rates only amounting to 11%.

Due to the increasing importance of social customer care, businesses increasingly automate simple requests from customers by developing and integrating chatbots on messaging platforms (York, 2017).

The possibility for developers to build chatbots on Facebook Messenger was introduced during Facebook's F8 event in April 2016 (Schlicht, 2016). Just seven months after the official presentation, there were already 34,000 chatbots available on the platform (O'Brien, 2016). But how is this fast diffusion possible? This essay makes the assumption that a platform's architectural principles lead to generativity which in turn enables the fast integration of chatbots. In order to defend this assumption, this essay adopts the case of Facebook Messenger and examines its architectural components in the light of the theoretical framework of layered modular architecture (LMA) proposed by Yoo et al. (2010). This results in the first research question:

RQ1: How does layered modular architecture explain generativity on Facebook Messenger?

The article examines the emerging implications of incorporating chatbots on Facebook Messenger. Chatbots present automated applications that interact with individuals, thereby enabling unprecedented ways of data and content generation. Hence, the automation potential generated by chatbots might have implications on generativity. This essay elaborates on this chain of thoughts by posing and answering a second research question:

RQ2: What are the implications of automation in the form of chatbots on generativity?

The article is structured as follows. Section II provides a review of the literature that is relevant for the considered phenomena. Section III presents the theoretical framework being utilised. Section IV presents the case of Facebook Messenger and subsequently analyses it through the lense of the adopted theoretical framework. Section V reveals contributions, limitations and future research directions.

Corresponding Author
Email Address: S.A.Draxinger@lse.ac.uk (S. Draxinger)

2. Literature Review

This section presents a review of the literature that is relevant to this phenomenon and focuses on the platform concept, the underlying architectural principles and perceived gaps in research about generativity.

Platform Concept

Over the last 20 years, ubiquitous digitalisation and concomitant innovation have deeply affected individuals' lives and organisational activities. Together with the emergence of the internet that was able to support any information service and with digital devices being able to gather, store and process different types of data, a fundamental phenomenon emerged: the digital platform (Tilson et al., 2010).

The platform concept has been researched by many scholars in different fields such as product development, technology strategy, industrial economics and information systems (Baldwin & Woodard, 2009). Since each research field examines platforms in different contexts, various definitions and terms emerged. In the product development field, Wheelwright and Clark (1992) coined the term "platform product". This refers to novel products that satisfy the needs of core customers but that can also simultaneously be modified into derivatives by the process of adding or removing features.

Technology strategists study principles of how to organise and manage the network of firms that surrounds platforms. Based on this research area, Gawer (2009, p. 45) introduces the term "industry platforms" and characterises them as "building blocks (they can be products, technologies or services) that act as a foundation upon which an array of firms (sometimes called a business ecosystem) can develop complementary products, technologies or services".

Scholars of economics literature adopted the term platform to describe a product, system, service or organisation that mediates transaction of goods, services or social currency between various agents, consequently creating value for all participants (Parker et al., 2016; Rochet & Tirole, 2003). Corresponding literature stresses the fact that network effects across different groups in multi-sided markets result in a 'chicken-or-egg problem' that has to be addressed by platform-owners by cross-subsidising between agents or following certain strategies (Parker et al., 2016).

Scholars of the information systems (IS) field such as Tilson et al. (2010) utilise the term of the "digital infrastructure" in order to characterise systems that can serve as platforms. Hanseth and Lyytinen (2010, p.1) use the term "information infrastructure" and specify it as a "shared, open, heterogenous and evolving sociotechnical system of information technology capabilities that are recursively composed of other infrastructures, platforms, applications and IT capabilities and controlled by emergent, distributed and episodic forms of control". The authors maintain that concepts such as platforms are used in relation with software-based systems, while terms such as digital or information infrastructures comprise a broader scope of systems from software to hardware infrastructures. However, the common denominator of platform-related literature in the information systems field, is to look at large and complex information systems as platforms, on which new products and services can be added to benefit from shared data (Tilson et al., 2010).

Despite different definitions of platforms across varying research areas, there are common structural features, that is, a common platform architecture which is based on the reutilisation and sharing of common core components across different products and services (Baldwin & Woodard, 2009).

Platform Architecture

According to Baldwin and Clark (2000), architecture is the function-to-component mapping with interfaces being incorporated into the design rules for building a modular system. Elaborating on the definition above, (Baldwin & Woodard, 2009, p.23) conclude that "all complex man-made systems, including all products and processes, have architectures". The authors further maintain that a crucial feature of platform architecture is the composition into components that stay fixed over a platform's life time and into components that vary across sections or which are modified over time. Besides elaborating on the reuse and sharing of core components, this definition proposes the importance of interfaces for platform-evolvement.

According to Baldwin (2008), interfaces constitute "thin crossing points" between platform components and serve as design rules that govern and constrain the relationships between components. Interdependencies and interactions, therefore have to comply with carefully designed interface specifications (Baldwin & Woodard, 2009). Additionally, interface creation is tightly related to the concept of modularity of physical products, since these points of control form boundaries between different modules of the architecture. Modules are platform-components, whose subcomponents are tightly interlinked with each other and loosely connected to other subcomponents of different modules (Baldwin & Clark, 2000). In a modular architecture, interfaces between different modules are highly standardised (Yoo et al., 2010). It allows a product to be decomposed into weakly coupled components that are linked by prespecified interfaces. This decreases complexity and increases flexibility in product design (Baldwin & Clark, 2000). Modularity itself describes the extent to which this decomposition is possible (Schilling, 2000).

Yoo et al. (2010) argue, however, that the ubiquitous digitisation – the process from translating analog signals into digital formats (Tilson et al., 2010) – and the growing incorporation of digital components into physical products, lead to a new form of product architecture: the layered modular architecture.

This new type is a hybrid between the modular architecture and the layered architecture that is characterised by the four weakly coupled layers of devices, networks, services and contents brought forth by digital technology. While components of a modular architecture are product specific and restricted by a fixed product boundary and thus following a functional design hierarchy (Clark, 1985; Baldwin & Clark, 2000), components of the LMA are 'product agnostic'. By connecting components pertaining to various layers, new products can be created (Clark, 1985). Furthermore, innovations can emerge at any layer, independent of other layers. LMA's characteristics are thus laying the grounds for generativity (Yoo et al., 2010).

According to Zittrain (2006, p. 1980), generativity is "a technology's overall capacity to unanticipated change through unfiltered contributions from broad and varied audiences". Tilson et al. (2010) describe generativity to be the prize for exploiting possibilities enabled by the flexible nature of digitising. In addition, the authors point to the paradox nature of control, which confronts the conflict between openness and closeness, to understand generativity and the dynamics of platform growth. Platform openness can be regarded as the degree to which a platform is centrally controlled to harness value and maintain power or to which extent it is open to the public (decentralised control) to benefit from efforts and innovations by third parties (Ghazawneh & Henfridsson, 2013). Platform openness can be studied through the lense of technical openness, that is, how Application Programming Interfaces (APIs) and Software Developer Kits (SDKs) restrict and control access to core components of a platform (Anvaari & Jansen, 2010). A further means to examine openness is by applying the perspective of organisational openness. This relates to ways in which platform-owners, developers and end-users are able to engage in development and usage of the platform (Economides & Katsamakas, 2006). The design of platform openness therefore, influences the degree of generativity and thus the potential of independent developers to innovate and produce content in the form of products or services on the platform (Eaton et al., 2015; Tilson et al., 2010).

After reviewing the relevant literature, one can assert that the notion of generativity occupies a prominent part in academia. However, there is little to no research about generativity of messaging platforms despite their increasing importance. Additionally, current definitions of generativity accentuate the manual participation of complementary providers. The increasing emergence of automation in the form of chatbots however, might have implications on generativity as we know it to date.

3. Theoretical Framework

Given the need to shed light on both the generativity of messaging platforms and the imposed implications on generativity by incorporation of automation in form of chatbots, the theoretical framework by Yoo et al. (2010) is adopted because of its holistic view of platform architecture and explanation of generativity as an ultimate outcome of these architectural principles.

The layered modular architecture constitutes a hybrid model between the modular architecture of physical products and the layered architecture of digital technology. Digital technology differs from traditional technologies in three distinct ways. First, it is reprogrammable, meaning that software code that manipulates data in specific ways can be changed without modifying the physical embodiment in which the data is stored and thus enabling a digital device to perform various tasks. Second, digitisation is translating heterogeneous, analog signals into machine-readable digital format, i.e. binary digits which can be processed by any digital device. This translation is commonly referred to as the homogenisation of data. Third, digital technologies enable digital innovation which is defined as new combinations of physical and digital elements. Increasing diffusion of innovation leads to positive network effects that further foster the creation of digital devices and services. This, again, enables digital innovation due to decreased entry barriers and learning costs. This cycle is referred to as self-reference. These features of digitisation mark a change in design practice and lead to the emergence of the layered architecture.

The layered architecture comprises four layers – contents, service, network and device. The device layer constitutes the physical machinery level (hardware) and the logical capability layer (operating system) while the network layer encompasses a physical transport layer (cables and transmitters) and a logical transmission layer (network standards and protocols). The service layer includes applications for the user (creating and consuming content) and the contents layer consists of data such as texts, images, metadata and directory information.

Most importantly though, the layered architecture highlights the separation between physical components of the device from the applications that run on it, that is, the disentanglement of the physical layer from the service layer which is enabled by the reprogrammability of digital products. Furthermore, the characteristic of data homogenisation enables the separation between the way data is stored and manipulated and the way it is transmitted which can be regarded as the disentanglement of the contents layer from the network layer.

The four layers constitute different design hierarchies and the respective component design on each layer is independent from other layers. This equips designers with combinatorial possibilities since they can put together components from different layers using standards and protocols (Gao & Iyer, 2006), thereby enabling great levels of generativity (Zittrain, 2006).

Due to the pervasive embeddedness of digital technology into traditional artefacts, the layered modular architecture arises. The degree to which the layered architecture extends the modular architecture determines the extent of generativity (Yoo et al., 2010). As was already noted in Section III, a full-blown modular architecture is restricted by a fixed product boundary, meaning that its elements adhere to a single design hierarchy (Clark 1985, Baldwin & Clark 2000), thereby rendering relationships between product and components nested and fixed. Components are product specific and the flexibility of a modular architecture is related to the substitution of elements within the single design hierarchy (Yoo et al., 2010). LMA on the other hand, is not based on a fixed product boundary, implying that its components do not follow a single design hierarchy of a given product but are regarded as elements that pertain to different layers with each layer following a different design hierarchy. Components can be loosely coupled, thereby inductively enacting a product (Clark, 1985). This means that a component can be regarded as a standalone product and simultaneously be used as an element in a different product in a variety of ways. The components are therefore product agnostic and their loose combination across layers enables the emergence of innovations on any layer thereby creating generativity (Yoo et al., 2010).

4. Case Analysis – Chatbots in the Facebook Messenger

This section presents the case of Facebook Messenger and its subsequent analysis through the lense of the theoretical framework, thereby answering the two research questions identified in Section I.

RQ1: How does layered modular architecture explain generativity on Facebook Messenger?

FM's architecture follows the principles identified by Yoo et al. (2010). The device layer consists of the physical machinery level (computer hardware) and the logical capability level (operating system). Facebook Messenger is built upon internal hardware such Central Processing Unit (CPU) and Random Access Memory (RAM). Concerning the logical capability, the Messenger can be downloaded on devices running on operating systems such as iOS, Android, Windows and Linux.

The network layer encompasses a physical transport layer (cables, transmitters etc.) and a logical transmission layer (network and application protocols). FM's data is sent in the form of bits via physical transportation means such as radio frequencies (e.g. Wi-Fi) and/or Ethernet cables. Furthermore, Messenger uses the Transmission Control Protocol/Internet Protocol (TCP/IP) as network protocol in order to provide a reliable, ordered and error-free delivery. The FM application utilises the Message Queue Telemetry Transport (MQTT) as application protocol (Karasiewicz, 2013) in order to send data to and retrieve data from the FM servers.

The service layer is concerned with application functionality. Here, users operate as they create, change, save and consume content. With FM, users can text, share photos and videos, voice call, videochat, use augmented reality, search for people that are on the platform and use and search for chatbots. People are also able to develop and deploy applications upon the Facebook Messenger Platform by using Facebook's SDKs and APIs (Facebook for developers, n.d.).

The content layer consists of data that is stored and shared and furthermore comprises metadata and directory information. Data bases take an essential role in this level. FM uses sophisticated data bases to store data such as texts, pictures and sounds, metadata and directory information about the origin, ownership and copyright of shared content. Components of FM's architecture are product agnostic and don't follow a single design hierarchy. Innovations and improvements in the physical machinery level such as CPU and RAM, do not have implications on developer tools or programming languages used to create applications on Messenger. Same counts for protocols such as TCP/IP and MQTT. Furthermore, alterations in data storage such as switching from relational databases to in-memory or column oriented databases don't affect the design hierarchy of other layers of the Facebook Messenger and vice versa.

FM serves as a standalone product but since its components are connected to different layers, they can be incorporated into other products as well, that is, Facebook Messenger as part of another product. Besides traditional texting (standalone product), many applications incorporate Facebook Messenger APIs in their architectures so that users can share content over Messenger with their friends (Facebook Messenger as component of a different product). Dubsmash for example allows its users to share content via Messenger (Trisha, 2014). The opportunity to bundle FM with a host of heterogenous devices such as smartphones, tablets, desktop computers, smart TVs and cars, equips designers with combinatorial possibilities since they can put together components from different layers using standards and protocols, thereby building new products. This implies that Facebook Messenger's designers cannot fully anticipate all the potential ways that Facebook Messenger as a component will be used. The layered modular architecture fosters generativity.

Chatbots represent the latest example of innovation that emerged on Facebook Messenger. They are being deployed within the application layer and the data they generate is stored in the contents layer. A chatbot can be built and incorporated in two different ways: either via the FM developer's guide or via an automated bot creator. While the latter increases the development speed, the first gives the developer more room for innovation (Schlicht, 2016). Chatbots are implemented via standard APIs (ibid.). CNN for instance built its own chatbot and thus acts and innovates on FM's service layer as a third party provider. CNN needs no specific product related knowledge about the Facebook Messenger. It simply has to include the API and use Facebook's SDK to connect their bot to the platform (Schlicht, 2016)

A layered modular architecture furthermore helps to scale up platforms quickly without creating too much complexity and thereby threatening performance and thus generativity. The instant messaging function can be regarded as Facebook Messenger's installed base. This is also commonly referred to as the core interaction, that is, the exchange of values that attracts most users to the platform (Parker et al., 2016). New features and applications are loosely added as peripheral components upon the core interaction through APIs and SDKs. Since they are not incorporated into the central core, they do not threaten speed and usability of the platform.

Chatbots are not integrated into the core of the platform. Users that do not wish to communicate with chatbots are thus not at risk to suffer from lower speed and platform performance because chatbots do not interfere or draw away resources from core activities, that is, instant messaging among users. A user simply has to use the search function of the FM to find and start a conversation with a chatbot. The implications of the increasing diffusion of chatbots lead to the second research question.

RQ2: What are the implications of automation in the form of chatbots on generativity?

Chatbots present automated applications, thereby enabling unprecedented ways of data and content generation. Chatbots take advantage of natural language processing (NLP) and machine learning algorithms to offer users a smarter and more engaging conversation experience (Anon., 2016). NLP enables the chatbot to extract intent and entities from a user's message. The intent represents the request or action the customer wants to take and entities offer information to provide context for a request or further information to finish an action. With users sending different messages, machine learning algorithms are used to identify similar phrases for each intent. Consequently, chatbots enable users to hold natural conversations to access contents and services so that they do not have to rely on traditional navigational and search capabilities (Galer, 2017). Chatbots then follow up with direct answers, requests for additional information or recommendations for possible actions (Anon., 2016).

A chatbot delivers data-driven results and because of its capability to learn over time, the process will get faster as the chatbot faces similar requests regularly (Newman, 2016). Additionally, chatbots should be able to learn within a conversation, that is, understand and elaborate on customer preferences to reply with tailored services or offers. After a user is starting a conversation with the bot, the intent and entities of the message are stored in the firm's database. The chatbot attempts to reply to the posed question or request in the best possible way. By reviewing the customer's reaction, the machine learning algorithm allows the bot to make changes to its perception about the customer and insert the updated data into the data base. In order to learn more about the customer, bots can use sophisticated analytics and data mining techniques to make sense of the gathered data and find patterns. This can result in intelligent follow-up messages that offer additional out of the box services which the customer generally wants but may not have thought of in that moment. This might increase customer satisfaction.

By determining what customers want most, chatbots might also unveil possibilities for new applications that help fulfilling those novel customer demands. Just like humans, the ability of chatbots to make meaningful connections with company-wide resources leads to success (Galer, 2017).

While generativity was generally perceived to be the outcome of creative thinking and innovation of third party providers, i.e., manual in nature, above reasoning strengthens the possibility of generativity becoming increasingly semi-automatic, with bots coming up with creative ideas and solutions and developers coding the proposed applications.

The aforementioned thoughts shed light on bot-utilisation in an external focus. A chatbot's internal applications might, however, be even more valuable. Chatbots have access to a firm's databases and thus are able to perform the same tasks as most applications in one integrated system. Chatbots could therefore be able to render applications obsolete in the near future. By deploying their learning algorithms, chatbots can understand and serve an employee's specific needs, thereby tailoring results and eliminating non-relevant data. This reduces the effort to use many applications and search through work data to complete daily tasks. Employees simply have to ask their individual chatbot to deliver the required information – no matter if internal firm related information or public information online (Newman, 2016). By accessing organisational information in real time, developers can allocate more capabilities to the creation of innovative and user-centric applications. Since more time is spent for development, this might increase the degree of generativity witnessed on a platform.

5. Conclusion

The essay examined the rise of chatbots on messaging platforms. The topic's relevance becomes clear in the light of the increasing diffusion and improvement of chatbot-technology and its implications on platform growth. A case study on Facebook Messenger has been presented and examined. The essay focused on two phenomena related to the architectural design, which enables the fast diffusion of chatbots, and to the potential consequences of chatbots on platform generativity. These two phenomena have been investigated and uncovered in the context of literature that is suitable for the considered topic. A theoretical framework based on the efforts of Yoo et al. (2010) has been applied to structure the examination of the case.

The essay conveys the following contributions. First, it gives evidence that FM's architecture adheres to the principles of layered modular architecture and thus gives explanation for the increasing emergence of chatbots on the Messenger Platform. Secondly, it shows that chatbots might be in the position to render generativity increasingly semi-automatic, since they are able – due to NLP and MP – to automatically gather and combine data in unprecedented ways, thereby creating new insights and ideas which are then manually codified into new applications by developers. The possibility of having totally automatic generativity is not unlikely, since chatbots might at some point possess the ability to write code themselves. Third, the essay points out how chatbots can increase the level of generativity, since they can serve as central applications which provide developers with tailored organisational information in real-time. Hence, developers have more time to create user-centric applications, thereby fostering generativity.

The present work also bears some limitations and future research directions. First, the essay examines a single case that is certainly relevant due to the size and importance of the company and its massive user base but might not cover all aspects and criteria important to the considered phenomena. Hence, research about other messaging platforms (e.g. WhatsApp, We Chat, Slack, etc.) might be enlightening in relation to the discussed issues. Second, the conceptual analysis is purely based on available sources and documentations and certainly lacks internal viewpoints. Thus, future works could give consideration to on-the-field interviews. Lastly, the essay focuses on chatbots which represent a novel technology. There is still a lack of academic literature about their implications and potential. Future research might therefore reveal further details about this technology.

References

Anon., 2016. LocalDesignSolutions. [Online] Available at: http://www.lds.com/coepost/enabling-a-smart-user-experience-using-chatbots/. [Accessed 31 March 2017].

Anvaari, M. & Jansen, S., 2010. Evaluating architectural openness in mobile software platforms. Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, pp. 85-92.

Baldwin, C. Y., 2008. Where do transactions come from? Modularity, transactions, and the boundaries of firms. Industrial and Corporate Change, 17(1), pp. 155-195.

Baldwin, C. Y. & Clark, K. B., 2000. Design rules. Volume 1. The power of modularity. Cambridge, Mass: MIT Press.

Baldwin, C. Y. & Woodard, C. J., 2009. The Architecture of Platforms: A Unified View. In: A. Gawer, ed. Platforms, Markets and Innovation. Cheltenham, UK and Northampton, US: Edward Elgar, pp. 19-44.

Clark, K. B., 1985. The interaction of design hierarchies and market concepts in technological evolution. Research Policy, 14(5), pp. 235-251.

Constine, J., 2016. TechCrunch. [Online] Available at: https://techcrunch.com/2016/07/20/one-billion-messengers/ [Accessed 8 April 2017].

Eaton, B., Elaluf-Calderwood, S., Sørensen, C. & Yoo, Y., 2015. Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. MIS Quarterly, March, 39(1), pp. 217-243.

Economides, N. and Katsamakas, E., 2006. Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. Management Science, July, 52(7), pp.1057-1071.

Facebook for developers, n.d. Facebook for developers. [Online] Available at: https://developers.facebook.com/docs/messenger-platform. [Accessed 7 March 2017].

Galer, S., 2017. Forbes.com. [Online] Available at: https://www.forbes.com/sites/sap/2017/02/28/what-to-do-about-the-chatbot-invasion-at-work/#6cb502815c59. [Accessed 29 March 2017].

Gao, L.S. and Iyer, B., 2006. Analyzing complementarities using software stacks for software industry acquisitions. Journal of Management Information Systems, 23(2), pp.119-147.

Gawer, A., 2009. Platform Dynamics and Strategies: From Products to Services. In: A. Gawer, ed. Platforms, Markets and Innovation. Cheltenham, UK and Northampton, US: Edward Elgar, pp. 45-57.

Ghazawneh, A. & Henfridsson, O., 2013. Balancing platform control and external contribution in third-party development: the boundary resources model. Information Systems Journal, March, 23(2), pp. 173-192.

Hanseth, O. & Lyytinen, K., 2010. Design theory for dynamic complexity in information infrastructures: the case of building internet. Journal of Information Technology, Volume 1, pp. 1-19.

Karasiewicz, C., 2013. IBM. [Online] Available at: https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_facebook_is_using_mqtt_on_mobile?lang=en. [Accessed 5 April 2017].

Newman, D., 2016. Forbes.com. [Online] Available at: https://www.forbes.com/sites/danielnewman/2016/06/28/how-chatbots-and-deep-learning-will-change-the-future-of-organizations/#473454be4734. [Accessed 1 April 2017].

O'Brien, C., 2016. VentureBeat. [Online] Available at: https://venturebeat.com/2016/11/11/facebook-messenger-chief-says-platforms-34000-chatbots-are-finally-improving-user-experience/. [Accessed 4 April 2017].

Parker, G. G., Van Alstyne, M. & Choudary, S. P., 2016. Platform revolution : how networked markets are transforming the economy - and how to make them work for you. New York: W.W. Norton & Company.

Rochet, J. C. & Tirole, J., 2003. PLATFORM COMPETITION IN TWO-SIDED MARKETS. Journal of the European Economic Association, 1(4), pp. 990-1029.

Schilling, M.A., 2000. Toward a general modular systems theory and its application to interfirm product modularity. Academy of management review, April ,25(2), pp.312-334.

Schlicht, M., 2016. Chatbots Magazine. [Online] Available at: https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca. [Accessed 6 March 2017].

The Q2 2016 Sprout Social Index, 2016. sproutsocial. [Online] Available at: http://sproutsocial.com/insights/data/q2-2016/. [Accessed 12 March 2017].

Tilson, D., Lyytinen, K. & Sørensen, C., 2010. Digital Infrastructures: The Missing IS Research Agenda. Information Systems Research, December, 21(4), pp. 748-759.

Trisha, 2014. TrishTech.com. [Online] Available at: http://www.trishtech.com/2014/12/dubsmash-record-share-videos-on-facebook-whatsapp/. [Accessed 3 April 2017].

Wheelwright, S. C. & Clark, K. B., 1992. Creating project plans to focus product development. Harvard business review, 70(2), pp. 70-82.

Yoo, Y., Henfridsson, O. & Lyytinen, K., 2010. Research Commentary —The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research. Information Systems Research, 21(4), pp. 724-735.

York, A., 2017. sproutsocial. [Online] Available at: http://sproutsocial.com/insights/social-media-trends/. [Accessed 12 March 2017].

Zittrain, J. L., 2006. The Generative Internet. Harvard Law Review, May, 119(7), pp. 1974-2040.