Research on Personal Intelligent Connected Services in Daily Life Scenarios (Postprint)

Authors: Guo Lu, Fan Wei

Date: 2023-10-08T00:00:00+00:00

Abstract

[Purpose / Significance] The rich and diverse APPs and intelligent hardware in daily life scenarios have brought new connection experiences to individuals. An analysis of personal intelligent connection services characterized by automation, proactivity, and personalization contributes to the integration of information services into new scenarios and the enrichment of service methods. [Method / Process] First, we summarize the main types of daily life scenarios, analyze the connotation and elements of scenarios, the connections among scenario elements, and data connectivity. Second, based on the connections among scenario elements, we discuss the switching between scenarios. Third, we propose constructing personal intelligent connections using "IF-THEN" logic, using "trigger-action" programming as the service implementation means, and analyze typical service applications. [Results / Conclusion] We propose a "trigger-action" programming-based personal intelligent connection service design method founded on "IF-THEN" logic, providing a reference for the design of personalized information services in daily life scenarios.

Full Text

Preamble

ChinaXiv Partner Journal: Academic Exploration

Research on Personal Smart Connection Service in Everyday Life Scenarios

Guo Lu, Fan Wei
School of Public Administration, Sichuan University, Chengdu 610064

Abstract

[**Purpose/Significance**] The rich variety of apps and smart hardware in everyday life scenarios creates new connection experiences for individuals. Analyzing personal smart connection services characterized by automation, proactivity, and personalization can facilitate the integration of new scenarios and diversification of means for information services. [**Method/Process**] First, we summarize the main types of everyday life scenarios, analyzing the connotation and elements of scenarios, the connections between scenario elements, and data connectivity. Second, based on the connections between scenario elements, we discuss switching between scenarios. Third, we propose constructing personal smart connections using "IF-THEN" logic, with "trigger-action" programming as the service implementation method, and analyze typical service applications. [**Result/Conclusion**] We propose a design method for trigger-action programming-based personal smart connection services grounded in "IF-THEN" logic, providing a reference for personalized information service design in everyday life scenarios.

Keywords: everyday life scenario; connection; IF-THEN; smart service
Classification Number: G252.61
Citation Format: Guo Lu, Fan Wei. Research on Personal Smart Connection Service in Everyday Life Scenarios [J/OL]. Knowledge Management Forum, 2020, 5(1): 36-46 [citation date]. http://www.kmf.ac.cn/p/198/.

1 Introduction

Everyday Life Information Seeking (ELIS) refers to activities where people acquire information not directly related to work tasks for self-positioning and problem-solving in daily life [**?**]. Wu Dan et al. summarized 26 categories of everyday life information themes corresponding to personal information acquisition behaviors in daily life, including social interaction, news reading, navigation, shopping, academic information queries, weather checks, and other aspects covering all facets of personal daily life scenarios [**?**]. Increasingly, scholars emphasize the important role of context in personal information behavior—any information behavior occurs within a context, which determines individuals' purposes and behaviors when using information systems [**?**]. Contexts in which everyday life information is acquired are called everyday life scenarios. According to the "2018 Scenario White Paper" released by the Internet Scenario Laboratory [**?**], individuals in everyday life scenarios use apps and smart hardware for reading, learning programming, recording social interactions, health diagnostics, consumption, and leisure activities. Vertical domain apps and smart hardware have greatly facilitated access to various segmented types of everyday life information, but excessive app installation and usage also burden users. Meanwhile, many needs and tasks require cross-application implementation—for example, saving numerous attachments from emails to cloud storage requires manually saving each attachment and then uploading it to the cloud, a process that is cumbersome and highly repetitive.

Creating cross-application connection services has been addressed in both academic research and industry applications. The most well-known connection service is the Internet of Things (IoT), which extends connected objects to any item based on the internet, forming a huge network for intelligent identification and management [**?**]. In recent years, the embedded services emerging in library and information science aim to connect librarians with users, users with resources, and users with environmental equipment, providing comprehensive, proactive knowledge services for users in library scenarios [**?**]. The connection concept has countless industry applications, such as Siri, Cortana, Google Assistant, and Alexa. Their technical principles mainly involve speech recognition and underlying application connections, converting user voice input into computer-readable commands and calling corresponding webpages, apps, or smart hardware on connected services to provide required services on a smart terminal. Smart home services aim to solve connection problems in home scenarios. IoT and 5G technologies have addressed basic "hard connection" issues, but there remains vast potential research space for "soft connection" issues that are omnipresent and continuous in ubiquitous information environments.

Users frequently switch between software applications in everyday life scenarios, and the cumbersome interface switching required for cross-application operations leads to a lack of fluidity in connection services. This is a widespread practical problem. How can we coordinate and call multiple apps and smart hardware to provide more efficient, seamlessly connected smart connection service experiences? Understanding personal users and examining their behavioral processes in everyday life scenarios, approaching smart connection services from the "scenario—connection—experience" perspective is the starting point of this research.

2 Analysis of Connections in Everyday Life Scenarios

Everyday life information seeking behavior does not occur in a vacuum but is closely related to the everyday life scenarios in which it occurs. To realize personal smart connection services in everyday life scenarios, we must first focus on everyday life scenarios themselves. Connections in everyday life scenarios include connections between scenario elements and connections between scenarios. Therefore, we first analyze the classification of scenario types to understand the connotation of everyday life scenarios. Second, based on this connotation, we identify the elements of everyday life scenarios. Finally, from the perspective of the information behavior process driven by personal needs, we analyze how scenario elements connect and how scenarios connect and switch.

2.1 Classification of Everyday Life Scenario Types

The classification of everyday life scenario types is related to the information environment. In early periods, individuals primarily used PCs to access information services. Since PCs cannot move, scenarios were relatively singular and less prone to change, resulting in fewer context-related studies. After the

emergence of Personal Digital Assistants (PDAs) with embedded sensors like barcode scanners and RFID, the number of perceivable scenarios increased. G. Chen and D. Kotz divided contexts into four categories: physical context, computational context, user context, and temporal context [**?**]. With the emergence and popularization of smartphones, apps have become increasingly abundant, and with the development of smart hardware, scenarios for acquiring everyday life information have also multiplied. H. Ko and C. Ramos [**?**] divided scenarios into home, hospital, pharmacy, company, restaurant, and meeting scenarios, but these are not standard scenarios—just some scenarios that systems can identify. Wu Sheng [**?**] divided scenarios into heavy scenarios and light scenarios. Heavy scenarios refer to those with high consumption, many users, and that have become lifestyles; light scenarios are relatively scarce compared to heavy scenarios.

After the rise of artificial intelligence, its application scenarios have become extremely broad, currently focusing on transportation, personal assistants, medical health, finance, security, education, and e-commerce retail, including specific applications like autonomous driving, voice assistants, and smart homes, covering all aspects of daily life [**?**]. Scenarios mentioned in AI applications are divided according to industries, with each scenario having a large scope that can be further subdivided into multiple sub-scenarios. Taking the Mijia smart home ecosystem's home scenario as an example, the home scenario is subdivided into coming-home, leaving-home, and home-office scenarios. Sub-scenarios are defined in two ways: one is through predefinition by Mijia of heavy scenarios with high usage frequency and many users for direct selection, such as coming-home and leaving-home scenarios; the other is through user customization based on personalized needs for scenarios with fewer users, such as pet-feeding scenarios.

Although scenario types are classified differently under different information environment and there is no unified standard, we can see that the connotation of scenarios is relatively stable, with the core being people and focusing on the things and objects around them. Scenarios exist objectively; what information and communication technologies can do is optimize the way scenarios are perceived, improve the convenience of perception, and cover more scenario types through software, hardware, and integrated information systems for comprehensive perception, subsequently generating computation, judgment, and application.

2.2 Identification of Everyday Life Scenario Elements

From the perspective of information service design, the connotation of everyday life scenarios identifies four elements: personal users, apps, smart hardware, and data. People are personal users; things and objects are apps and smart hardware; what connects personal users, apps, and smart hardware is data.

Personal users select apps and smart hardware based on their needs. Apps can directly provide users with service content data obtained through networks,

collect data from scenarios, or remotely connect to and control smart hardware. Smart hardware can directly provide services to users, be remotely controlled by apps, or collect data from scenarios through sensors. Data includes personal data, physical environment data, and service content data. We discuss these four elements separately:

(1) **Personal Users.** Users in everyday life information seeking research include children and adolescents, retired women, pregnant women, the elderly, immigrants, international students, low-income individuals, and university teachers and students [**?**]. The users in this study are not subdivided in the current research stage but refer to individual users who can normally use computers, smart terminals, apps, and various services in daily life. Future research will conduct specialized studies on different types of populations according to actual needs.

(2) **Apps.** Software is an abstract concept with different forms and references. In mobile internet and IoT environments, apps on mobile terminals have become synonymous with software. App categories cover all aspects of daily life, and individual users increasingly use and depend on apps in everyday life. Apps can identify scenarios through user input data and behavioral logs, such as querying free classrooms to identify school scenarios. They can also control smart hardware, such as controlling air purifiers and smart refrigerators through the Mijia app, or obtain service content data needed by users.

(3) **Smart Hardware.** Smart hardware includes sensing devices (such as RFID, iBeacon), wearable devices (such as smart bracelets, VR glasses), and smart home devices (such as smart refrigerators, smart locks). Smart hardware plays three main roles in scenarios: first, directly providing services to users, such as smart air conditioners cooling; second, embedded sensors collecting personal physiological data, behavioral data, and physical environment data in scenarios and transmitting it to apps for active scenario perception, such as temperature sensors in smart air conditioners collecting room temperature and transmitting it to the Mijia app for scenario judgment; third, being connected and controlled by apps.

(4) **Data.** Data in scenarios includes personal data, physical environment data, and service content data. Personal data is divided into physiological and behavioral data collected by smart hardware, and input data and behavioral logs collected by apps. Physical environment data is environmental data collected through smart hardware, such as location and temperature. Service content data refers to scenario-corresponding service content obtained from networks through user input or other scenario perception methods, such as when a user inputs “schedule query” to perceive a school scenario, and the app obtains school course schedule-related content from the network.

2.3 Analysis of Scenario Element Connections and Scenario Switching

To enable personal users to use apps and smart hardware in everyday life scenarios more closely aligned with personal needs and behavioral processes, we must analyze personal daily needs and behavioral processes, combined with the usage patterns of scenario elements, to explore how elements in scenarios should cooperate.

Personal users may need multiple apps and smart hardware to fulfill a complete need, thus requiring clicking on multiple apps and operating multiple smart hardware during the behavioral process to achieve personal needs. For example, to understand top news from multiple news apps requires opening each app separately to read.

In everyday life scenarios, personal needs serve as the original driving force. [Figure 1: see original paper] shows the connections between scenario elements. Connected elements include personal users and apps, personal users and smart hardware, apps and apps, apps and smart hardware, and smart hardware and smart hardware (the latter achievable through apps that connect and control smart hardware). Connections are divided into non-data connections already completed in real scenarios and incomplete data connections. In Figure 1, dashed lines represent non-data connections already realized in real-life scenarios, so scenario connections need to focus on the data connections represented by solid lines, requiring the establishment of underlying data channels to enable data flow and connect scenario elements. Data flows bidirectionally between personal users and apps and between apps and apps, while flowing unidirectionally between personal users and smart hardware and between apps and smart hardware.

Each scenario has scenario elements, with different apps, smart hardware, and data constituting different scenarios. While connecting scenario elements can realize connections within a single scenario, to provide users with more seamless and fluid information service experiences in everyday life scenarios, we must not only achieve connections within single scenarios but also realize connections between scenarios and free switching. Scenario switching occurs because elements in scenarios change, such as personal location movement or temperature rise. Individuals use different apps and smart hardware in different scenarios, generating different data. Therefore, to achieve free switching between scenarios, the connection of scenario elements and data flow is the foundational architecture. On this basis, we also need to focus on what data enables scenario switching, between which apps and smart hardware data flows, and then quickly call the corresponding apps and smart hardware to achieve connections and seamless switching between scenarios.

3 “IF-THEN” Logic Analysis

There are many logics for connecting scenario elements and between scenarios. To select logic more aligned with personal needs, we must consider personal

behavioral habits. Through M. J. Bates' Berry Picking model [**?**], we can see that personal needs and behaviors in everyday life scenarios are dynamic and changing, not single-linear, and most personal information behaviors are selective, phased, and sequential. A. K. Dey et al., through the visual programming system iCAP, found that when individuals are asked to use context-aware applications, they often specify operations in "IF-THEN" logic [**?**]. "IF-THEN" is a conditional judgment logic containing "IF condition" and "THEN action," including selection of trigger conditions and actions, and the stages and sequence of "trigger" and "action," which aligns with the information behavior patterns mentioned in the Berry Picking model. Therefore, we choose "IF-THEN" logic to connect scenarios.

3.1 Composition of "IF-THEN" Logic

The "IF this THEN that" statement configured using "IF-THEN" logic is a complete "IF-THEN" logical connection (Connection). It has three essential elements: Trigger, Action, and Service [**?**], as shown in [Figure 2: see original paper]:

(1) **Trigger.** The trigger in "IF-THEN" logic corresponds to "this" in "IF this THEN that"—the "trigger condition." When an element in the scenario where the individual is located meets the trigger condition, it automatically triggers the corresponding "IF-THEN" logical connection. Triggers can be any scenario element. Hardware sensor components can serve as triggers, and systems composed of multiple smart hardware and software can implement trigger solutions, typified by Apple's iBeacon and Google's Eddystone, both location-based services (LBS) that trigger corresponding services when the corresponding platform approaches iBeacon or Eddystone. For example, setting up iBeacon in a library pushes potentially interesting books to users when they enter. Apps can also serve as triggers, such as IF receiving a meeting notification email THEN add to calendar, where the email box is the trigger.

Triggers can also be divided into active triggers and passive triggers. Active triggers monitor in real-time and execute actions when trigger conditions are met, such as IF room temperature exceeds 30 degrees THEN start air conditioning, which monitors room temperature in real-time. Passive triggers refer to triggers activated after an individual performs an operation, such as IF bookmarking a webpage THEN save to Evernote, where bookmarking the webpage is the passive trigger.

(2) **Action.** Action corresponds to "that" in "IF this THEN that"—what is done by other services when trigger conditions are met. Everything triggered after "THEN" that satisfies trigger conditions is an action. Actions and triggers have a many-to-many relationship, and triggers and actions are not absolute but determined according to personal need settings. Actions were originally tasks that required personal users to judge conditions

and execute manually, but by configuring a complete "IF-THEN" logical connection, the platform calls the service's API to proactively execute this action in appropriate scenarios, avoiding cumbersome operations, achieving seamless and automated services, and saving time and improving efficiency.

(3) **Service.** Both triggers and actions require apps or smart hardware as carriers, collectively called services. Triggers and actions are specific data and functions of certain parameter indicators in services, such as "receiving email" being a function of email, making email a service. Using services, countless "IF-THEN" logical connections can be created, and by setting execution frequency, "trigger-action" smart services in personal everyday life scenarios can be realized.

3.2 Scenario Switching Based on "IF-THEN" Logic

"IF-THEN" logic can help personal users freely enter scenarios, leave scenarios, and switch scenarios. Free entry into a scenario means that when entering a scenario, when an element in the scenario meets trigger conditions, tasks in that scenario are initiated. Free exit from a scenario means that when leaving a scenario, tasks in that scenario are automatically ended. Free scenario switching means that the trigger and action services in one "IF-THEN" logical connection belong to different scenarios, achieving free switching between scenarios.

In the processes of individuals entering, leaving, and switching scenarios, coordinating apps and smart hardware, automating tasks, and providing proactive services can deliver seamless, fluid, simple, and convenient everyday life information services. Coordinating apps and smart hardware means that on the basis of connecting apps and smart hardware, "trigger-action" programming coordinates apps and smart hardware to complete tasks, including coordinating app-to-app, app-to-smart hardware, and smart hardware-to-smart hardware. By automatically executing actions when trigger conditions are met, manual operations are avoided and task automation is achieved. Through active triggers monitoring in real-time, proactive services are provided when trigger conditions are met.

For example, based on a personal user's basic daily behavioral route—buying household items in the morning, leaving home, driving to work, and picking up children after work—we enumerate some "IF-THEN" logical connections in corresponding home, travel, office, and school scenarios to illustrate how to freely connect personal users, apps, and smart hardware within and between scenarios according to personal daily life needs, achieving these aspects of smart connection services, as shown in .

3.3 "Trigger-Action" Programming for "IF-THEN" Logic

By setting "IF-THEN" logical connections, individuals associate triggers with actions to automatically execute actions when trigger conditions are met in

specific scenarios, implementing a programming model in the form of "IF trigger, THEN action"—what we call "trigger-action programming" (TAP) [**?**]. A study by B. Ur et al. found that trigger-action programming can express most behaviors [**?**]. Trigger-action programming is a simple programming model that enables individuals to easily create rules for implementing smart connection services.

Trigger-action programming is a type of end-user programming (EUP) [**?**], which refers to activities and tools that allow end users (non-professional software developers) to program computers. The most common end-user programming is spreadsheets. Spreadsheet macros provide individuals with interfaces for customizing batch processing—by "recording macros" to record a series of operations to be executed as a macro, executing this macro when needed can achieve one-click repetition of multiple operations, simple and convenient. Trigger-action programming allows individuals to program through touch control interfaces, easily creating complex tasks that even inexperienced individuals can quickly learn to use.

Trigger-action programming allows defining abstract "IF-THEN" logic between independent technologies/brands, organizing triggers and actions in hierarchical structures to allow selection between different abstraction levels. It supports multiple triggers and actions, where triggers can be connected through "OR" logical operators and actions can be connected through "AND" logical operators [**?**]. In practice, the selection of "OR" and "AND" logical connection operators has no rigid rules and needs to be determined according to specific scenarios. Additionally, G. Desolda et al., in defining similar triggers and actions, specify that up to three additional constraints can be designated: Who, Where, and What [**?**].

Trigger-action programming based on "IF-THEN" logic uses RESTful APIs to create data pipelines between services, then obtains service permissions through OAuth 2.0 (Open Authorization), creating the foundation for individuals to freely connect services. RESTful API is essentially a resource-oriented software architecture based on the HTTP protocol. When designing, all content on the network is abstracted as a resource, with each resource having a unique identifier (URI). Operations on resources are actually achieved by operating URIs through commands like GET, POST, PUT, and DELETE. Through RESTful APIs, data format connections between different ecosystem products can be achieved, solving data compatibility issues between different apps and smart hardware in different ecosystems.

The OAuth 2.0 mechanism can provide certain protection for user privacy and security. OAuth 2.0 generates a short-term token that allows third parties to access user data within limited permission scopes. Tokens are short-term and automatically expire, and users can also revoke tokens at any time to render them invalid [**?**]. By configuring "IF-THEN" logic on the backend, the data and functions of each service become triggers and actions. On the user interface, individuals only need to perform simple selections of triggers, actions, and ser-

vices to customize scenarios, and can use already written "IF-THEN" logic to achieve "trigger-action" programming.

Smart connection services that adopt trigger-action programming on the basis of connecting scenario elements through "IF-THEN" logic are called trigger-action programming-based smart connection services. There are already many such services based on "IF-THEN" logic, represented by Yahoo Pipes, Microsoft Flow, Zapier, and IFTTT. We now proceed with case analysis to summarize the advantages of each service.

4 Case Analysis of Trigger-Action Programming-Based Personal Smart Connection Services

Foreign countries already have some implemented applications of trigger-action programming-based personal smart connection services that have formed healthy ecosystems. In contrast, there are currently no successful application cases in China. We analyze relevant foreign services, summarize their advantages and disadvantages, and provide references for domestic information service providers in designing personal smart connection services.

4.1 Yahoo Pipes

In 2007, Yahoo launched a micro-pipeline aggregation tool called Yahoo Pipes to aggregate various feeds. Yahoo Pipes provided a graphical interface allowing individuals to set rules according to their needs to aggregate information from different sources and publish it. Technology expert T. O'Reilly called "Yahoo Pipes a milestone in Internet history" [**?**].

Although Yahoo Pipes already provided a graphical interface, its difficulty of use was still relatively high—this was one of the main reasons leading to Yahoo Pipes' shutdown. However, Yahoo Pipes' ideas of automatically obtaining Web resources, aggregating Web services, and reusing rules influenced the later design of trigger-action programming-based smart connection services.

4.2 Zapier

As apps in office scenarios proliferated rapidly but did not communicate data with each other, and individuals frequently needed to perform repetitive work across different apps, Wade Foster et al. launched Zapier in 2012 for office scenarios to automate repetitive tasks for staff and save working time [**?**]. Unlike Yahoo Pipes' complex operation interface, users only need simple clicks to connect two or more apps, build their own app workflows, and automatically execute repetitive tasks.

In Zapier, services are called Apps, a complete "IF-THEN" logical connection is called a Zap, and actions in a Zap are extended to one or more actions or searches. An element in Zapier is called a Task, with each piece of data run through Zap counting as a Task. For example, if Zap automatically adds

100 emails to Dropbox, it executes 100 Tasks. Each Task executed by Zap is something that originally required manual execution by individuals.

4.3 Microsoft Flow

Leveraging its advantage of having numerous office software products, Microsoft launched Microsoft Flow in 2016, also applicable to office scenarios. Prior to this, most services under Microsoft and some other office software operated in their own silos. If multiple services needed to be used, data transmission was a challenge [**?**]. Microsoft Flow provides an automation platform to solve data transmission difficulties, allowing individuals to connect different online services and local platforms, such as Office 365 and SQL Server. Microsoft Flow allows individuals to create automated workflows between apps, achieving automated processes + tasks for services like obtaining notifications, synchronizing files, and collecting data [**?**].

Services in Microsoft Flow are called Connectors. A complete "IF-THEN" logical connection is called a Flow, extending actions to one or more. In addition to "IF-THEN," Flow also provides "IF NOT" logic.

4.4 IFTTT

IFTTT is a typical trigger-action programming-based service applied to everyday life scenarios. It breaks the isolated situation of apps and smart hardware on the Internet, unifying connection methods through APIs so that individuals can freely combine apps and smart hardware like in real life, more aligned with people's usage habits. IFTTT extends "IF-THEN" logic, changing the original "IF this THEN That" to "IF this, THEN that, AND that" or "IF this, AND that, THEN that," allowing individual users to select multiple triggers and multiple actions.

IFTTT also has the concept of ingredients, which refer to single data fragments returned from triggers each time a small program runs. Small programs can collect various elements provided by triggers and use them to execute actions. IFTTT has been widely applied, winning the Best Accessibility Experience award in the 2017 Google Play Best Apps [**?**].

4.5 Comparative Analysis of Trigger-Action Programming Services

Since Yahoo Pipes has shut down, we conduct comparative analysis among Microsoft Flow, Zapier, and IFTTT. All three belong to trigger-action programming, with the common goal of providing non-developers with a simple connection method to connect apps and smart hardware and achieve cross-platform automated services. They share some common characteristics, such as having supported services, triggers, and actions, and allowing individuals to combine tasks by configuring triggers and actions [**?**]. We now introduce and compare them from aspects of background, positioning, applicable scenarios, scenario connections, and logic, as shown in .

Based on the above analysis, in terms of connection logic, in addition to IF-THEN, we can also extend to IF NOT, IF-AND-THEN, IF-THEN-AND, and other logics. In terms of supported apps and smart hardware, we should make them as rich as possible, covering multiple scenarios in users' daily lives to provide seamless, personalized, proactive information services and realize personal smart connection services in everyday life scenarios. Regarding user operation interface, Yahoo Pipes' high difficulty of use was a main reason for its shutdown, so when designing services, we should simplify the operation interface for individual users as much as possible.

5 Summary

Personal smart connection services in everyday life scenarios mainly consist of four components: scenario element connections, connections and switching between scenarios, "IF-THEN" logical connections, and trigger-action programming implementation. We summarize as follows:

5.1 Scenario Element Connections

First, we must determine detailed types of personal users' everyday life scenarios. Only with specific usage scenarios can products and services attract users. We propose four common types, but other specific types require detailed analysis based on individual users' conditions. Second, we propose four elements in scenarios: personal users, apps, smart hardware, and data, and analyze their connection methods, noting that data connections require attention to data flow direction.

5.2 Connections and Switching Between Scenarios

On the foundational architecture of scenario element connections, we also need to achieve connections between scenarios. Different scenarios consist of different apps and smart hardware, requiring analysis of which apps and smart hardware to simultaneously call to achieve connections and switching between scenarios.

5.3 Defining "IF-THEN" Logical Connections

There are many logics for connecting scenario elements, with "IF-THEN" logic being the most intuitive. When defining "IF-THEN" logical connections, we can predefine some commonly used "IF-THEN" logical connections in individuals' everyday life scenarios for direct selection. We should also provide personal customization functions, supporting multiple triggers, multiple actions, and IF NOT logic.

5.4 Designing Trigger-Action Programming

Trigger-action programming is a simple and effective implementation approach for "IF-THEN" logic. Using RESTful APIs to connect various services and

OAuth to obtain service authorization enables data flow between services. By hiding complex "IF-THEN" logic in the backend and only providing individual users with simple operation interfaces where they only need to click to select triggers and actions, automatic configuration of "IF-THEN" logical connections can be completed, achieving personal smart connection services in everyday life scenarios.

To advance the integration and deepening of everyday life information service research into new scenarios, we summarize several levels that need to be connected and implementation approaches in designing personal smart connection services for everyday life scenarios. Our main research contribution is clarifying the approach for designing personal smart connection services, including classifying everyday life scenarios; identifying everyday life scenario elements (personal users, apps, smart hardware, and data); analyzing element connections within scenarios and connections and switching between scenarios; and technically implementing scenario element connections and inter-scenario connections. We propose "IF-THEN" logic that conforms to personal everyday life information behavior to construct connections and propose simple and easy-to-use trigger-action programming as an implementation approach.

We have theoretically explored the design of personal smart connection services, which awaits further practical testing through service implementation. Realizing personal smart connection services in everyday life scenarios requires clarifying many detailed issues, such as subdividing types of everyday life scenarios and obtaining permissions for apps and smart hardware.

References

[1] SAVOLAINEN R. Everyday life information seeking: approaching information seeking in the context of "way of life" [J]. Library & information science research, 1995, 17(3):259-294.

[2] WU Dan, LIANG Shaobo. A review of research on network information search behavior in multi-device environments [J]. Journal of Library Science in China, 2015, 41(6):109-127.

[3] BAWDEN D. The turn: integration of information seeking and information retrieval in context [J]. Journal of documentation, 2005, 43(2):821-822.

[4] Scenario Laboratory. 2018 Scenario White Paper [EB/OL]. [2019-11-11]. http://www.199it.com/archives/734109.html.

[5] LIU Qiang, CUI Li, CHEN Haiming. Key technologies and applications of the Internet of Things [J]. Computer Science, 2010, 37(6):1-4.

[6] ZHOU Xiaojie, LIU Haixin, ZHANG Chunyang. A review of embedded service research in Chinese libraries [J]. Library Science Research, 2012(12):19-22.

[7] CHEN G, KOTZ D. A survey of context-aware mobile computing research [R]. Technical Report TR2000-381. Hanover: Dept. of Computer Science, Dartmouth College, 2000.

[8] KO H, RAMOS C. A survey of context classification for intelligent systems research for Ambient Intelligence [C]// Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on. Washington: IEEE, 2010: 746-751.

[9] WU Sheng. Scenario Revolution [M]. Beijing: Machinery Industry Press, 2015:197-220.

[10] China Industry Information Network. Analysis of China's Artificial Intelligence Application Fields in 2016 [EB/OL]. [2019-11-11]. http://www.chyxx.com/industry/201612/482350.html.

[11] China Internet Network Information Center. The 44th Statistical Report on China's Internet Development [EB/OL]. [2019-11-11]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201908/P020190830356787490958.pdf.

[12] XIAO Yongying, HE Lanman. Progress in foreign everyday life information seeking behavior research (2001-2010) [J]. Library and Information Service, 2012, 56(5):112-118.

[13] BATES M J. The design of browsing and berrypicking techniques for the online search interface [J]. Online review, 1989, 13(5): 407-424.

[14] DEY A K, SOHN T, STRENG S, et al. iCAP: interactive context-aware tool programming [C]// Proceedings of the 4th International conference on pervasive computing. Berlin: Springer, 2006:254-271.

[15] HOY M B. If this then that: an introduction to automated task services [J]. Medical reference services quarterly, 2015, 34(1): 98-103.

[16] GHIANI G, MANCA M, PATERNO F, et al. Individualization of context-dependent applications through trigger-action rules [J]. ACM transactions on computer-human interaction, 2017, 24(2):1-33.

[17] HUANG J, CAKMAK M. Supporting mental model accuracy in trigger-action programming [C]// Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. New York: ACM, 2015: 215-225.

[18] UR B, MCMANUS E, HO M P Y, et al. Practical trigger-action programming in the smart home [C]//Proceedings of the SIGCHI conference on human factors in computing systems. New York: ACM, 2014: 803-812.

[19] Wikipedia. End-user development [EB/OL]. [2019-11-11]. https://en.wikipedia.org/wiki/End-user_{development}.

[20] CORNO F, DE RUSSIS L, ROFFARELLO A M. A semantic web approach to simplifying trigger-action programming in the IoT [J]. Computer,

2017, 50(11): 18-24.

[21] DESOLDA G, ARDITO C, MATERA M. Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools [J]. ACM transactions on computer-human interaction, 2017, 24(2): 12.

[22] PARECKI A. OAuth 2.0 [EB/OL]. [2020-01-26]. https://oauth.net/2/.

[23] O'REILLY T. Pipes and filters for the Internet [EB/OL]. [2019-11-11]. http://radar.oreilly.com/2007/02/pipes-and-filters-for-the-inte.html.

[24] Zapier. What is Zapier? [EB/OL]. [2019-11-11]. https://zapier.com/learn/getting-started-guide/what-is-zapier/.

[25] WEARE K. Microsoft flow reaches general availability [EB/OL].[2019-11-11]. https://www.infoq.com/news/2016/11/microsoft-flow-ga.

[26] Microsoft Flow. Automated processes + tasks [EB/OL].[2019-11-11]. https://flow.microsoft.com/zh-cn/.

[27] Google Play. The winners of the 2017 Google Play Awards [EB/OL]. [2019-11-11]. https://blog.google/products/google-play/winners-2017-google-play-awards-are/.

[28] RAHMATI A, FERNANDES E, JUNG J, et al. IFTTT vs. Zapier: a comparative study of trigger-action programming frameworks [J/OL]. arXiv:1709.02788, 2017[2020-01-20]. https://ui.adsabs.harvard.edu/abs/2017arXiv170902788R.

Author Contributions

Guo Lu: Responsible for data collection and analysis, initial draft writing.
Fan Wei: Responsible for topic selection, framework design, guidance and revision.

Note: Figure translations are in progress. See original paper for figures.