A Comprehensive Literature Review on Approaches, Techniques & Challenges of Mashup Development

Urooba Sehar, Iqra Ghazal, Hamna Mansoor, Sehrish Saba

Abstract— After revolutionary invention of Web 2.0 in field of World Wide Web, Mashup development has got a lot of attention of software practitioners and researchers as an End-user Development (EuD) application area. In this era of emerging technologies in field of EuD, researchers are exploring different aspects of Mashup development to help individuals and enterprises build effective and efficient mashups and mashup development tools for themselves to enhance the performance of their business processes and to cop up with evolving requirements of their businesses and at individual level for making EuD effective, user friendly and feasible specifically for novice and non-technical users. Having a huge amount of researches that are being conducted and published in last few years, when an enterprise or individual decides to move towards adoption of Mashup development they face a lot of difficulty while collecting required information of approaches and techniques for their business specific requirements. Main purpose of this study is to build a broader leel understanding related to different aspects of Mashup Development by presenting current state of literature on the approaches and techniques for the Mashup development and challenges being faced by practitioners regarding Mashup development in order to make users aware of effective approaches that they can adopt for themselves in accordance to their needs. In this paper, we have presented a comprehensive literature review based study exploring the approaches and techniques that are being proposed by different researchers for Mashup development. We have explored the literature from the past decade and analyzed the findings of researchers in field of mashup development. This study also has a detailed discussion on the challenges that are faced by practitioners in process of implementation of the mashups development and in use of various mashup tools.

Index Terms— End-user Development, End-user Programming, Mashup Approaches, Mashup Development, Mashup Development Challanges, Mashup Techniques, Mashup Development Frameworks.

IJSER

1 INTRODUCTION

IN this age of technological reforms, End-user Development (EuD) is not anymore an unknown term to the practitioners in field of software engineering and information technology. After revolution of web 2.0, mashups have received a lot of hype among researchers that is an emerging application of EuD for Web application development and other business applications. It is gaining recognition on a wide scale among software researchers and practitioners. The term mashup is derived from the music industry that is about mixing music songs for creation of a new version of those songs despite the source of the music. The software practitioners have grabbed that term "mashup" for combing various services into a one single web application or a service [1]. A Mashup itself is usually web application that has capability of combining the data from different data sources into a single joint place. Content or data that is used in mashups is mostly collected from any third party with the help of a public interface like Application Programming Interface (API). Mashups have the capability for the users to combine the data and services together at one place from different web applications for creating a new integrated tool that is able to fulfil the requirements of the users. Aim of mashups is to provide rapid development by End Users which having less programming knowledge. Mashups are commonly used for situational purposes it mainly combines specific required data and functionality from different resources to fulfil user needs. Different frameworks are proposed till now which helps to create mashups for non-professionals but all frame works required some skills to operate them in this paper author compare all the frame works which are used in mashups development approaches tell their limitations as they give no knowledge about selecting data sources and were very time consuming. It is observed that mashups development tools follow four interacting styles including webpage customization, wire paradigm mashups, spreadsheet mashups and programming by demonstration interactive style but they are not as simple to handle by non-programmers they still need some basic information and background to work on mashup development tools. In the last few decades, a wide range of mashups approaches have been proposed by various researchers with claim of simplifying the process of building different types of mashups so that every user who are non-technical and novice users become able to create mashups efficiently and effectively for them without having need of any training sessions or any type of technical knowledge.

At this stage, we can see a tremendous amount of research that is conducted in field of EuD, more specifically now researchers are exploring different aspects of Mashup development. The present researches are focused on concepts like approaches for mashup development, techniques that could be used for the development of mashups, to employ the amazing uses and advantages of mashups that can be utilized in different fields of life, various tools that are being developed for the mashup development for individuals and enterprises and the challenges that are faced by end users while using the mashup development tools and developers for the development of the tools and for the implementation and adoption of mashup development. The difficulty that is faced by practitioners is to

find the correct and relevant information degrading all the aspects of mashup development while adoption of mashups for their enterprise and domain specific requirements. Main purpose of this study is to provide a detailed and comprehensive study having all the required information related to mashup development adoption at one place with a concise and clear understanding.

This study has presented a detailed analysis of the present state of literature on the approaches and techniques for the Mashup development and challenges being faced by practitioners regarding Mashup development in order to make users aware of effective approaches that they can adopt for themselves in accordance with their requirements. We have studied and analyzed the prior literature from the past decade and analyzed the findings of researchers in field of mashup development. This study also has a detailed discussion on the challenges that are faced by practitioners in process of implementation of the mashups development and in use of various mashup tools.

The structure of the remaining paper is as follows: Section 2 of the paper has a discussion on the review of prior studies on mashups development techniques, approaches and challenges. Next, Section 3 has description of our research methodology, including the process for collecting the data and literature about challenges, techniques and approaches presented by the researchers as well as the design of the paper. Section 4,5,6,7 includes systematic and extensive discussion on different well known and effective approaches and techniques for Mashup development as well the challenges being faced. In Section 8, we have discussed and provided our analysis and summery of mashup development approaches and techniques and we have also discussed how the challenges can be overcome that are discussed in the Section 4. Lastly, Section 9 of the paper that concludes the paper summarizing our study and presenting the shortcomings and gaps of the current state of literature.

2 Literature Review

In this section, we have a stated a detailed discussion on present literature work of various researcher to build an understanding of how much and about what aspects of Mashup development research has already being conducted, what are their findings, what are shortcomings of the present state of research and identified the research gaps that still need attention of researchers.

Mashup development as general application of End-user Development is being discussed by many researchers to have a generic understanding about functionality of mashups. In a study by Grammel et al. [2] they have provided a survey on different mashup development environments. They have explored features provided by different mashup development environments to facilitate end users to develop mashup of different types for themselves. Kulathuramaiyer [3] has presented a study with detailed discussion on mashup development and its role in this emerging technological world. In a paper by Yu et al. [4], they have selected various commonly used mashup tools and discussed how those tools are able to facilitate the development of web applications. Their paper focuses on identifying emerging characteristics and dimensions which we can use to compare and analyze tools and approaches. It also explains the difference between mashup creating tools and other compositions. It also explains the benefits of mashups developments tools by comparing them traditional. They have discussed about environment, frameworks and approaches help to develop mashups. In their study they also put focus on pros and cons of multiple approaches under some attributes.

Some of researchers have discussed the techniques for mashup development. Some of the techniques proposed are specific to different fields of life and some are generic that could be used for nay field. Researchers also have proposed techniques to cop up with the challenges that are faced while mashup development. Patel et al. [5] has presented a comprehensive study on the principles, concepts and techniques related to Mashup Development by exploring different mashup tools from point of view of software engineering for EuD. They have selected and reviewed ten different mashup tools for their research work and have also provided a categorization of mashup tools in their study. In this study an emerging technique is introduced for mashup boosting purpose it has user-friendly instructions which require no or less professional support because they are using "open" APIs, AJAX and RSS as a group of interrelated web development methods. Their proposed technique helps end users for creating, uploading, sharing and mixing of content through diverse data integration from required sources. Hummer et al. [6] has introduced a debugging technique for declarative development of data mashups. Web 2.0 is used to visualize intermediate results in each processing which keep record of its steps if there is any error it can locate where is the problem occurs. The implementation is based upon Web aggregation. This process helps for concrete corrections and improvements. Ravallion [7] has presented a study related to mashup indices of development. As composite index for which present state of the theory and practices does not provide enough guidance for its design so their study focuses on useful new composite indices of development will be required in order to make the theory catches up with measurement practices of the field. Edberg et al. [8] has proposed a study focusing on the currently used mashup techniques for maintenance and have found that most of them composed of parts and chunks from various formal methodologies.

Cappiello et al. [9] has also presented a study in which they have discussed the importance of mashup development and its increasing demand. But according author there are some areas which are neglected, most particularly the issues related to quality aspects for such applications. This paper covers the dimensions that discusses the intrinsic quality factor of the mashup components. Furthermore, authors have proposed a technique which they call the assisted composition process in which quality is given priority, when users are given recommendation regarding the completion of the mashups. Scotch et al. [10] has conducted a study on application of mashup de-

velopment tools in field of Public health. They have described how mashup development can be used in this field and how can it contribute towards public health informatics. Minhas et al. [11] have carried out the research to find out the different tools and techniques the assist in the mashup development. Liu et al. [12] have provided an architecture that converts the typical Service Oriented Architecture into a Mashup based SOA which put emphasis on integrating multiple web services through APIs into a desired service needed by the end user on runtime without much knowledge of implementation. Xu et al. [13] has proposed a platform architecture on the basis of Services Oriented Architecture (SOA). Along with this, they have also presented a SLA-driven service management framework for enhancing the reliability, and also for the dynamic adaption of the mashup application used in the platform.

Many researchers have proposed many approaches for the mashup development for novice users. Jiang et al. [14] has introduced an approach for hybrid open API selection to develop Mashups for developers to efficiently and accurately find the open APIs that they need by using approach of both discovering and recommending APIs. They have also evaluated the effectiveness of their proposed framework using different evaluation approaches on the basis of real data. Fischer et al. [15] has presented multiple approaches and limitations of web mashups. In their study they emphasis on the challenges of semi-automated tools which are not efficient and possess lack of quality data integration mechanism from multiple sources over the web. The study tried to work on automated mashup creation with useful framework keeping the focus on user interests to help them achieve their needs efficiently. In this paper framework is suggested which automatically combine data from resources to create mashups and user model that store user knowledge and his needs according to his skills which help him to generate mashups for individual user. Web Mashups are emerging technology to combine heterogeneous and versatile data from different sources to make one for own need. Complex mashups required much data and technical support which face many challenges but this paper shows technological changes od development and maintaining mashups i.e., data integrity. Complex mashups take more processing steps and are hard to debug if could not gain what was expected. Li et al. [16] has presented an approach in field of mashup development for API recommendation using mashups. They have proposed a probabilistic model for assisting mashup developers by recommending them a list of Application Programming Interfaces that can be used for compose any required mashup by giving the descriptions and details of the mashup. Anarfi et al. [17] has presented a Reinforcement Learning based approach for mashup development that is capable of adaption of dynamic nature of web API quality properties for the recommendation of web Application Programming Interfaces for optimal mashup solution. Their proposed approach also has capability of recommending replacement web APIs to the mashups that already exist in a dynamic environment, where the quality properties of the component web APIs are continuously changing and evolving. Gu et al. [18] has proposed a mashup development approach for recommendation of service package using text description mining. They have evaluated their proposed approach on real-world dataset. Results of their study show that the proposed approach is capable of achieving a higher level of accuracy and also has better results in comparison with other approaches.

Zhao et al. [19] has also presented on-the-fly web based mashup approach for better usability and user satisfaction for multiple web source integration. In their study, the component model dynamically separates the business service and interface of user and a browser based prototype model is also provided for demonstration. Albinola et al. [20] has presented innovative approach called mash light for mashup creation using web 2.0 widgets. Mash light is light weight and can run on browser without any server requirements. In their study, easy composition of web services is emphasized with no technical expertise. Aghaee et al. [21] has conducted a research that defines the design and implementation of EUD approach for these two requirements first is satisfying the needs of non-programmers according to their abilities and the second is to enable creation of any type of mashup by using manual approaches. Furthermore, it also provides an overview and classification of available existing approaches of EUD. It also pays importance that understanding end user mindset is necessary to give them specific natural and understanding tool for mashup development. Cheung et al. [22] has proposed a mashup approach for biomedical data. The main focus of their study is get benefits for data integration within health care and life sciences of using Semantic Web technologies like mashups. J. Jeffrey Hanson [23] has study with discussion about the mashup creation approaches and techniques. In his book he has presented that 3 categories user interface, data and process are needed to create a mashup of any style and technique. Paredes-Valverde et al. [24] has presented the study of mashup creation environment, programming languages, methods and approaches. In their study they emphasis on semantic technology that understand what is best according to the situation to accomplish user needs. This study has presented to help companies to choose best fit of tools and approaches as well as researchers to grasp the state and need of mashup in emerging technology field. Troudi et al. [25] proposed a novel data extraction and event sensing strategy that differentiates from earlier approaches by including the mashup notion. Their method may extract data from a variety of social media sites with varying architectures. It aims to detect occurrences based on three primary dimensions: subject, time, and place.

As end users without having knowledge of technology are getting benefited by mashup development tools and techniques. There are still many challenges that researchers and practitioners face in field of mashup development. It sometimes requires technical knowledge to some extent with limits the user of mashup development. Many researchers have discussed the challenges that are being faced in mashup development field. By the increase use of mashup solutions mainly in organizations context it raised questions of high-quality operations. In a paper by Majer et al. [26] operational challenges are discussed that mashup structure is improved by the

awareness of operator teams about their contribution to meet the end user needs and give high quality services. Failures are reduced which were happening because of uncoordinated change management mechanism within the CMDB and structural overview. The decomposition of SLAs into quantifiable parameters is a good approach for improving mashup solutions. Gnanasekar [27] has presented an architecture to create, reuse and manage script based service oriented mashups which focuses on dynamic discovery of web APIs using quality of service for improved results. In their study, they have tried to eliminate the effort of end users in the development phase for learning code and selection strategies of APIs from multiple sources. Huang et al. [28] has a study that focuses on the system that saves time and effort and increases usability by understanding the situation and context of the user and then create itself widgets for mashup without any manual settings.

The system dynamically provides mashup widgets to create mashup on runtime by keeping an eye on user needs. In a paper by Koschmider et al. [29] generic definition of mashup is defined and how it differs from the technology SOA. Challenges are discussed which we face in mashups development. Many languages are proposed for the construction of mashups, among all them this paper proposed Orc language as most developed and documented language. This paper also questioned methodical issues and sketched best guide for the construction of mashups. They have illustrated systematic instruct for separating and defining mashups and other existing systems for combining sources. In their study they also discuss about multiple platforms, languages and challenges of mashups with classification of mashups under some criterion. According to the authors mashup programming is the latest trend but this is in the infancy yet, we need framework and tools for the development of mashups. Goussis et al. [30] has carried out a study for detailed discussion about the important factors that shall be considered while development of the mashup tools. They have also discussed the challenges that are being faced by the mashup developers for the development of mashup creators. Daniel et al. [31] also has a published research on challenges of mashup development and have reviewed different enterprise mashup tools. They have identified two different scenarios that are different in terms of heterogeneity of services to be combined, the diverse nature of user needs, and the sophistication of the actors and the tools supporting their work.

From literature review, it is very evident that a huge amount of research is being conducted in field of mashup development. Hence, there is still need to have a detailed study that has collection and comparison of approaches and that are present in the literatures and also the challenges that are faced by practitioners.

3 Research Mehtodology

In this section, we have discussed the research methodology of our paper. We have adopted a literature review based approach to conduct this study. First step of the research was to gather information related to our three core research focuses including approaches of mashup development, techniques that are being proposed for them and the challenges that creators and users face while adoption and creation of the mashups. Second step of the research was to analyze the information gathered from different sources and classify them info different categories of the approaches and similarly for techniques and challenges. Next step of the study was to document and write the relevant information that was gathered while analyzing the present state of literature into their appropriate place and part. Last step of the research is to summarize and discuss the collected data and information and to discuss the approaches and their example along with the challenges that were associated with them. In following section, we have discussed in detail the step of our research methodology.

3.1 Data Collection

First step of carrying out this study was to collect the data about the approaches and techniques proposed by different researchers and also the researches that have discussed the challenges that creators and practitioners face while adoption and creation of the mashups. For this phase, we have visited different online research journal and conference paper libraries and directories like Science Direct IEEE Explore, Research gate etc. and have searched the journal papers that were related to our research and then selected the papers from list of available papers depending on the nature of research and the data that was being included in the researches.

3.2 Anlyzing Information

Second phase of the research was to analyze and read all the research papers that we have selected during the first stage of the research. In this phase we have gathered and classified the information depending on the type and usage of approach, technique or challenge. This is the phase of reading different papers and gathering the useful and required information and writing the information down as initial drafts of the paper.

3.3 Documenting the Collected Information

Third step of the study was to document all the collected information and writing down the approaches and techniques that we have studied while reading different research papers. Similarly, we have written down the challenges that we have studied from the current state of research.

3.4 Summerizing the Results and Findings of the Study

Last step of this study involves the summarization and colculsion of all the gathered information at one place that was being documented and written into different sections and parts so that it all comes to make sense to the readers and users that are interested in adoption of the mashups. This summarization contains the results of our study containing the categories and sub categories of techniques, approaches and challenges that were identified and proposed by different researches in past few years.

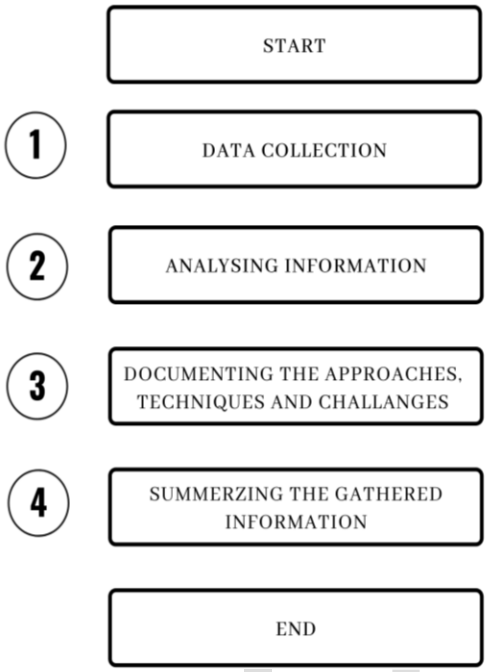

Fig. 1. A pictorial depiction of steps involved in our proposed research methodology for our conducted literature review study for analysis of approaches, techniques and challenges of mashup development.

4 MAHSUP APPROACHES CLASSIFICATION

In following section, we have discussed various types of approaches for mashups that can used by creators on the basis of the requirements of the mashup they are going to create. Following part has described some of the classes that mashup approaches are divided into.

4.1 Media Mashup Approaches

Media mashup approaches are those that are used for rejoining and merging the digital media files that contain any of the following elements i.e. texts, images, audios, videos, and animations. Its goal is to split each data type into its own presentation on the same page. The ability to display several results on a web page is provided by media mashup. As a result, users may see multiple data connected to the same query, moreover the relationships between various data sources, such as whether data is being maintained on the same web host.

Example. One example of media mashup approaches is to create a map displaying highly ranked high school features directly on a map. Google maps and a list of high schools with high ranking is used for this purpose.

4.2 Web Mashup Approaches

Web mashups are web applications that are created by combining and repurposing data and functions that are available on the Internet. Mashups may be made with traditional web programming languages, special mashup tools like IBM's mashup tools, or dashboard systems like Netvibes. In the same way, the Mashup idea is used in the business world. Despite the growing popularity of mashups, robust development tools and frameworks are strong and adaptable, but they are only appropriate for programmers, and mashing up a new application usually requires a large amount of manual programming.

4.3 Manual Mashup Approaches

To combine data sources, make visualizations, and expand capacity, users must have programming or scripting expertise.

4.4 Semi Auto Mashup Approaches

It guides users through the process of creating a mashup application using the tools supplied. It is further divided into the following categories.

1 Spreadsheet Oriented

Amico calc and MashSheet are two examples of this category, in which users enter data directly into a spreadsheet.

2 Widget Based

Using a visual editor, allow users to construct the mashup. Yahoo Pipes and Intel Mash Maker are two examples of mashups in this area.

3 Demo Based

Allow users to mash up their data by offering examples and a visually step-by-step procedure for accomplishing the data integration job. Dapper and Karma are examples of this category.

4.5 Auto Mashup Approaches

It enables the development of mashups without the participation of the user, since the tool chooses and invokes the resources (data, visualization, and operation) automatically.

4.6 Mashup Building Environment Approaches

1 Consumer Mashups Approaches

A consumer mashup is a browser-based program that mixes resources (such as architecture or material) from many public or private sites and arranges them using a straightforward web user interface.

Example. Dapper, Grazr, DERI pipes, Intel mash maker, Microsoft Popfly and Openkapow.

2 Enterprise Mashups Approaches

An enterprise mashup combines several assets (e.g., dynamic data functionality) from numerous channels. In position to react to their aims, these mashups mix data and application features from other systems, such as ERP, CRM, or SCM. Enterprise mashups necessitate the concern of privacy, accountability, and business standards. Enterprise mashups eliminate the need for a mediator by quickly combining and expressing public and private company's assets from various resources.

Example. Apatar, Data mashups, IBM InfoSphere MahsupHub, JackBe Presto, Procession, Serena mashup suit, RssBus, Snap Logic and TIBCO PageBus

IJSER

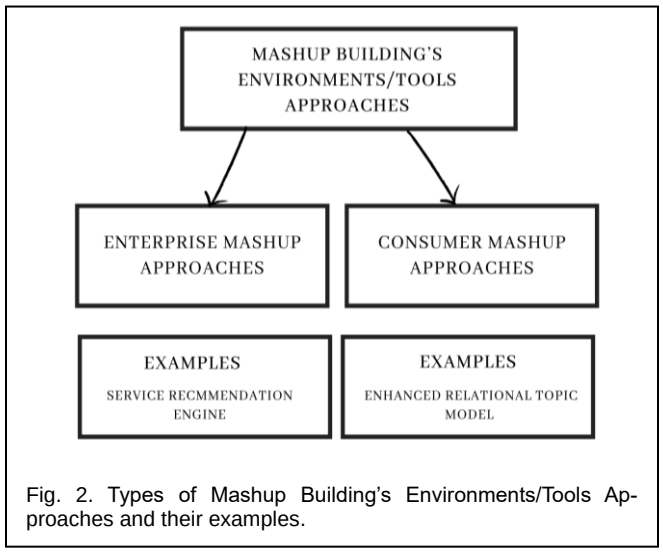

Fig. 2. Types of Mashup Building's Environments/Tools Approaches and their examples.

4.7 Mashup Modality Approaches

1 Extraction Mashups

The extraction mashup may be understood as just a data wrapper that gathers materials from several sites before combining them into a single content page.

Examples. Dapper, Grazr, IBM InfoSphere MahsupHub, Openkapow and Rssbus.

2 Flow Mashups

A flow mashup is able to customize the flow of resources for a Web page by integrating materials from many sources. Within the mashup application, the resources are changed and merged.

Examples. Apatar, Data mashups, DERI pipes, Intel mash maker, Microsoft Popfly, JackBe Presto, Procession, Serena mashup suit, Snap Logic, TIBCO PageBus and Yahoo! Pipes.

4.8 Database Based Approaches

Database Based mashup approach is an approach that is used for the creation web mashups along with support for database-level integration and which allows mashup to creates fo working with a standard abstract model while still having access to database systems' strong capabilities directly.

4.9 Service Based Approaches

A method for transforming old systems into service-oriented mashups. They emphasize modelling the business requirement up front, then examining existing legacy systems to determine whether such capability exists and can be included into the mashup, according to their paradigm. "The incremental modelling process is as follows: (a) if a business requirement can be fulfilled by one of the existing legacy elements, simply wrap it using the Quality of service attributes; (b) if there is a difference between the existing legacy element and the requirement, and the gap can be filled throughout service wrapping, then accustom the legacy element into a new service; (c) if the gap cannot be filled, then create a new service for the requirement."

4.10 Passive Approaches

It focuses on creating plugins that operate with the user's existing browser, watch what they're looking at, and offer relevant sites for possible mashing.

4.11 Poractive Approaches

When the mashup process grows more complex (for example, process modelling or advanced interface integration), it becomes required. It allows users to create mashups in complex integration domains by first providing samples of what the end mashup should look like; the tool for this sort of mashup then attempts to imitate the format of the end result, allowing mashups to be created by non-programmers.

4.12 Mashup Asset Integration Approaches

1 Presentation Mashups

A presentation mashup is concerned with obtaining information and layout from many Web sources while ignoring the underlying data and application functionality. Prebuilt widgets are easily dragged and dropped into a common user interface for this form of mashup. The construction of a presentation mashup usually involves little or no programming experience.

Examples. Data mashups, Dapper, Grazr, Intel mash maker, Microsoft Popfly, Openkapow and Rssbus.

2 Data Mashups

A data mashup combines data from several sources (e.g., Web applications, tweets, or simple HTML) into a single page layout (for example, integrating various services to gather the weather predictions, events, and images for a certain city). The user combines data from several sources and personalizes the flow of information of a Website page including data from various sources, for instance.

Examples. Apatar, Data mashups, DERI pipes, IBM InfoSphere MahsupHub, JackBe Presto, Microsoft Popfly, Procession, Serena mashup suit, Snap Logic, TIBCO PageBus and Yahoo! Pipes.

3 Functionality Mashups

A functionality mashup is a system that integrates data and application features from multiple providers into just one. APIs are used to retrieve the functions.

Examples. Examples of functionality mashups include JackBe Presto, Serena mashup suit and Procession.

4 Mapping Mashups

Mapping-Mashups ate mashups that are used for making mixture of data in form of maps, e.g. Google maps, photo-/Video-Mashups like combination of data, e.g. from Flickr, Search/Shopping-Mashups like connectivity of processes for contrasting product prices into Internet sites, and News-Mashups are some examples of mashups based on the assets being that are being blended togather. [29]

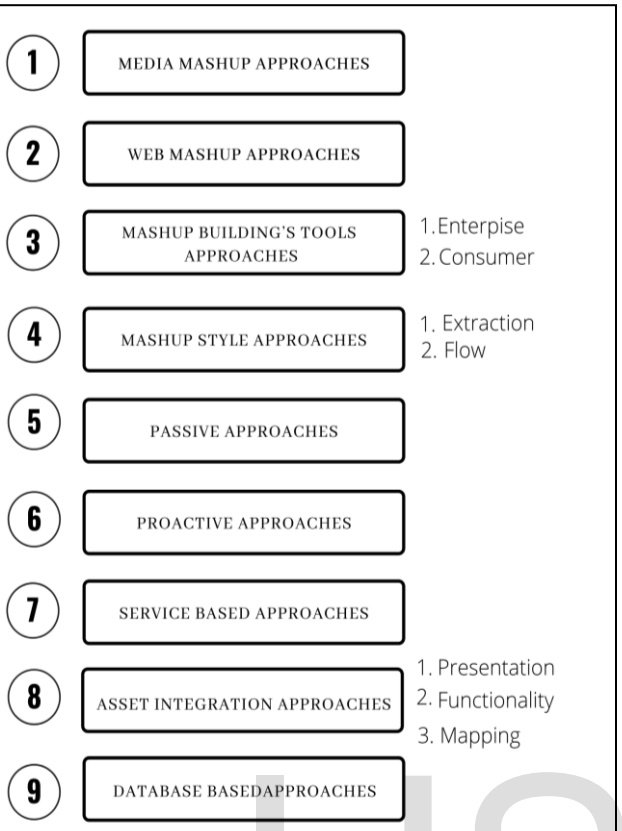

Fig. 3. Classification of Mashup development approaches with respect to their use for different types of fields of life and approaches.

5 Mashup Development Approaches

In this section, we have discussed in detail various approaches for development of mashup of different types including data mashups, enterprise mashups, consumer mashups and data integration related to mashup that different researchers have proposed in their published research work.

5.1 E-RTM Model for API Recommendation

Enhanced Relational Topic model is a topic-based approach for API recommendation for mashups that has implementation in R language. Inventor has built new model on the basis of basic RTM model for improving the results of API recommendation. Li et al. has proposed this approach for incorporating the annotation relationship between services and tags and the composition relationship between mashups and APIs to discover the latent topics. Along with the fame of APIs that we have considered for recommendation and also for characterizing the preferences of the users towards APIs along with the functional consideration. On the basis of the relevance of the topic and also how popular an APIs is, the changes of existing relationship and association between mashups and APIs is evaluated and related APIs are then for building a mashup that is required. They have collected a service data set from the website ProgrammableWeb.com and have a discussion on the statistical properties of the current mashups and APIs. As last stage of their study, they have also conducted the experiments on a data set. The results of their study show the effectiveness of their proposed method as comparison to other approaches that are being proposed. [16]

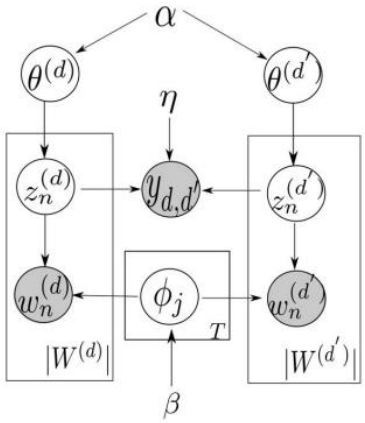

Fig. 4. Enhanced Relational Topic model two document segment.

5.2 Social Media Mashup Approach

It is a novel method to meeting the demands of social media users and enabling them to be better informed about news and current events. [25]

1 Construction

Using APIs to retrieve data from numerous social media sites with varying architectures, such as Twitter, Facebook, YouTube, and others. It focuses on multiple sorts of data, such as words, photos, and videos. The goal then is to extract real-world occurrences in greater detail from the recovered data using three fundamental dimensions: subject, time, and place.

2 Data Extraction

Twitter, YouTube, and Google Plus are among the options. This decision was taken to guarantee that various sorts of data, such as words, photos, and videos, are used. It chooses four broad disciplines to cover a wide range of activities, including news, technology, sports, and entertainment. It selects four legitimate accounts of traditional media in each social media for each domain. For Twitter, it utilizes an API Rest and Twitter4j API, for Google Plus, it uses a Google API, and for YouTube, it uses a YouTube API. It creates xml files from the collected user profiles, which form the corpus utilized in the following phase.

3 Event Fetching

To detect events, it suggests a hybrid technique. It's a hybrid of retrospective and online detection techniques. The method for detecting events is based on three fundamental dimensions, known as basic dimensions, which include the topic, time, and place. This method uses files containing user profiles that were retrieved in the previous phase as input and produces a list of recognized events as output.

4 Topic Fetching

This component's goal is to find tags that indicate the core theme that explains the occurrences. In this stage, it will use Twitter, a social media site that performs other social media in terms of giving timely news. The goal of this phase is to find Hash Tags, which are important words that start with "#" and are used to designate the topic of tweets.

IJSER

5 Component Fetching

The first phase is to find the components that better characterize the primary tag detected from each social media account using the co-occurrence technique, and the second step is to identify the tags that relate to locations or nations. To pinpoint the exact location of occurrences, it employs a gazetteer, which is essentially a geographic dictionary.

6 Media Mashup

After analyzing the primary theme and characterizing components from each social media event, it uses a digital mashup to mix and integrate the identified multimedia components for each event, resulting in a new description of events that is more detailed and includes a variety of data types.

In a nutshell, the suggested method is a hybrid strategy that combines retrospective and online detection methods. This combination allows to recognize previously occurring events, follow the course of the identified events, and be quickly notified of any new occurrences.

5.3 Textual Description Mashup Approach

A mashup development approach is proposed by Gu et al. [18] for recommendation of service package using text description mining. Their proposed approach is capable of recommending a package of services which can be used for the development of a new mashup that is required. They have conducted an experimental study and shown the results of their study show that the approach is capable of There are two main parts of their approach.

1. Service Recommendation Engine

This part of the framework receives the description of the mashup in textual forma and then later it translates that text for analyzing the specification of the required mashup. A component of it breakdown the text into Elementary Discourse Units (EDUs), on the basis of which the parser links the discourse tree of EDUs. Then the recommend the list of services for all the EDUs of the tree.

2. Service Repository

It is a container of the history of large number of mashups. On the basis of discourse tree analysis and LSI (Latent Semantic Indexing) model, this approach can generate the relation matrix of services.

Hence this approach then recommends the relevant service packages on the basis of analysis of the discourse tree. Fig 2. shows the example of discourse tree.

5.4 On the Fly Approach

There is no distinction between design-time and run-time in this component model. Each component is fully functional at runtime after it has been loaded. As a result, developers may use it to certify components. Developers may alter components on the fly if they didn't meet the criteria. The UI and service business logic are separated using this component paradigm. Furthermore, the approach defers to developers the flexibility to customize the UI structure and appearance. In a nutshell, developers may qualify and adapt components in real time.

Developers may construct components at runtime, and services can be triggered just-in-time since components are constantly at runtime. Composition UI is structured similarly to component UI. As a result, composition may be done on-the-fly since component adaptation can be done concurrently.

Example. In the case of a weather map scenario, developers should first utilize the searcher panel to find and extract the three components from the environment. Once the components have been received, they will be initialized and shown. Following that, the developer can connect with these components for qualifying (e.g., looking for weather forecasts for various cities). If components do not totally meet requirements, they can be adjusted using the configuration panel. The UI will update quickly as developers submit new UI settings for on-the-fly adaption. Qualification on-the-fly can be done at the same time as composition. Once the connection is established, position markers, for example, will appear immediately. Developers can detect and correct any geo-data inconsistencies in real time. The components can then be assembled by developers.

5.5 Automatic Generation Mashup Approach

Automatic mashup applications creations are nowadays being very interesting in research field as an evolutionary approach for creating the mashups. The research community has just lately begun to take an interest in the autonomous production of mashup apps. The framework is designed for mashups that include non-web functionalities (e.g., widgets). a method for creating mashups that is both adaptive and automated SOAP-ful or RESTful knowledge web services are automatically constructed via semantic web service composition based on a user's interests, duties, and experience in this mashup architecture. The gathered information is combined and displayed as a mashup. Smart mashups are the focus of a variety of academic projects. MashMaker is a browser plugin that adds mashup capability to the standard surfing experience. MashMaker does not need users to specify the mashup in advance, as workflow-like technologies do.

Additional examples are PiggyBank and MaxMash that works as automatic construction of mashups.

5.6 Automatic Generation Mashup Approach

In a data-oriented mashup, data from one or more externally hosted sites is combined in an application window or web page. Data is returned from a site or server in the form of XML, RSS, JSON, Atom, and other formats in this sort of mashup. Much of the information returned comes from data-oriented services, also known as Data-as-a-Service (DaaS), which don't need any third-party procedures between the service provider and the service customer.

1 In-Process Data Mashup

Data is mixed together in-process data-oriented mashups utilizing application or browser technologies like JavaScript, AJAX, or the Document Object Model (DOM). Data from one or more sites/servers is obtained and utilized as values or configuration settings to create user interface artefacts within an application process or browser page in this type of mashup.

2 Our-Process Data Mashup

Data is mixed together in out-of-process data-oriented mashups utilizing technologies like Java, Python, Perl, C++, XML, and XSLT, to mention a few. Within a server-side process or a separate client-side process, data is obtained from

one or more sites/servers and utilized as values or configuration settings to construct new data models.

5.7 Process Based Technological Mashup Approach

Frameworks like JEE,.NET, and Ruby on Rails can be used in mashups for business applications or online applications. Inter-process/inter-thread communication mechanisms like as shared memory, message queues/buses, remote procedure calls (RPC), and others are used to integrate functionality in a process-oriented mashup. Despite the fact that data is transferred across processes and threads, the end result is different from a data-oriented Mashup in that it aims to develop new data models.

5.8 Other Mashup Approach

1 Programming Paradigm Mashup Approach

A variety of technologies use an application programming interface to generate mashups, this manual construction is only done by professional developers. IBM WebSphere is a dynamic web software design and implementation environment. It facilitates the reuse of online services as well as the quick combination of different web services.

2 Scripting Language Mashup Approach

It appears that creating such scripts in a timely manner is too difficult for a non-coder, because more complex mashups would need a significant quantity of quite complex script code.

Examples. Some of examples of scriprting languages based mashup approaches are Google mashup editor and Web mashup scripting Language etc.

3 Spreadsheet-Based Mashup Approach

Spreadsheet-based technologies concentrate on data recombining. The data is immediately put into a spreadsheet, unlike with wire-oriented technologies. This implies that a data source's results are pushed to cells that the user has previously chosen. Following that, the cell values are used as inputs for data source searches.

Examples. are StrikeIron SOA Express and Extensio Excel Extender etc.

4 Wiring Paradigm Mashup Approach.

Wire-based tools that use a visual wiring of fundamental building elements to blend and integrate data, feature, or appearance. Wiring or piping of various units, connector, elements, or blocks is another term for this manual connectivity. The accessible components serve various functions (e.g., data fetching, data processing, data display, etc.) and must be linked in order for the mashups to work together. These environments use RESTful data sources.

Examples. are Apatar, IBM Damia, Marmite, SABRE, JackBe Presto Wires, Microsoft Popfly, Yahoo Pipes and Openkapow etc.

5 Programming by Demonstration Approach

Demonstration programming allows users to learn a system by providing experiences. Users with little programming abilities can utilize the Internet Scrapbook system to automate repetitive surfing operations. The user can select parts from several web pages that he finds interesting and combine them into a customized mashed-up site. The data is extracted using the HTML structure of the web page in question. Karma uses demonstration programming to pull collections of data from web sites using easy drag-and-drop of web page components. The system makes use of the browser's DOM tree data to derive a data table. By matching identifier and value pairs, the data may be automatically combined with other records produced from other sites.

Additional examples are Dapper and Potluck that uses APIs and URLs and works on semantics. [15]

6 Hybrid Scripting Language Approach

It is based on the type shifting paradigm, it promotes unity. Type morphing, like inheritance in object-oriented contexts, is the language's ability to cast any primitive type to another when appropriate. Data is transformed among heterogeneous kinds according to a set of established criteria. Users can progressively overload (or mash) model components using an interactive interface until a satisfactory solution is found. [32]

7 Hybrid Open Selection of API Approach

There is a rapid increase in interest in the mashup development in the field of software engineering. Authors have focused to use the existing open API`s to complete the requirement of mashup developers. For this purpose, they have proposed a hybrid open API approach for the selection of API for the mashup development. Their proposed model has two approaches one is the user story driven open API selection approach and another one is the multidimensional-information-matrix based open API selection approach. In the user story based approach, first it selects 3 components from the story and then extract the three properties from the API. Similarity calculation is then performed on both. In the second approach the mashup, the API and the invoking relationship are first stored then the factorization machine model is applied to calculate the scores between the mashup and the API. [14]

6 Mashup Development Techniques

This section contains summary of the mashup techniques that are being proposed by different researchers in recent years for development of mashups for applications in different fields of life.

6.1 Techniques for on the fly approach

On Fly Service technique is a techniques used for the development of web based mashups. Technique involves the following steps for development of the web mashups. An example of on fly service techniques of mashup development is demonstrated below.

In the case of a weather map scenario, developers should first utilize the searcher panel to find and extract the three components from the environment. Once the components have been received, they will be initialized and shown. Following that, the developer can connect with these components for qualifying (e.g., looking for weather forecasts for various cities). If components do not totally meet requirements, they can be adjusted using the configuration panel. The UI will update quickly as developers submit new UI settings for on-the-fly adaption. Qualification on-the-fly can be done at the same time as composition. Once the connection is established, position markers, for example, will appear immediately.

1 Component Retrievel

JavaScript, a dynamic scripting language, is used to create the composition environment, which is hosted in a web browser. As a result, the environment is able to load WBS component definitions without having to restart. A WBS component may also be created without compilation and deployment once its specification is downloaded, because it is implemented by JavaScript. A component is at runtime after it has been instantiated. It connects to a service that is already operating and can respond to user inputs with full functionality.

2 Composition of a Programming Interface.

Three connectors are used. The simple connector is the first sort of connector, and it allows two components to contact directly. This connection allows components to have a one-to-many publisher/subscriber relationship. That is, one component publishes an event, which is then subscribed to by other components. Event listeners define the publisher/subscriber connection. Each listener has an event publisher, an event type, an event subscriber, and a subscribing component function.

The data format mismatch can be fixed using the second data connector. Developers may become upset if they can only link two components at a time while trying to design a complicated flow with more than two components. As a result, the third type of connector available in our composition environment is the flow connector. To enable a certain type of flow, each flow connector corresponds to an integration-pattern, such as "Aggregator" or "Process Manager." The flow connectors are built as a specific form of WBS component with blank user interfaces and programming interfaces that link to other components.

3 Construction of the User Interface

When developers put WBS components together, their UI elements are combined into a new UI. The structure and display setup of this new UI must then be adjusted by developers. The setup of the new UI is identical to that of the atomic component UI. Furthermore, in order to facilitate on-the-fly UI construction, environment monitors changes to the UI's configuration. When a new configuration is set, it is applied to the environment right away, and the new UI is presented. As a consequence, developers may get instant feedback on assembly outcomes.

6.2 Techniques for Presentation-Based Approach

1 Putting Presentation Artifacts Together

The simplest kind of presentation-oriented mashup is the aggregation of UI artefacts within a web page in a portal-like manner—that is, employing distinct regions within a single HTML page to keep them totally separated from one another. UI artefacts including gadgets, widgets, HTML snippets, JavaScript inclusions, and on-demand JavaScript are embedded in an HTML document utilizing layout elements and methods like HTML tables and CSS positioning in this paradigm.

2 Combining Data for Presentations

A browser page may also create and change user interface artefacts by combining data from many sources in formats including XML, RSS, Atom, and JSON. The data is obtained from one or more sites, processed by the browser, and UI artefacts are built or modified using scripting techniques like DOM manipulation to change the final HTML document.

3 Using the Exml Http Request Object and Ajax

Asynchronous JavaScript and XML (AJAX) is a combination of tools and strategies for dealing with data feeds and building dynamic websites with a complex look and feel using JavaScript. The use of JavaScript and CSS methods to change a UI component on a web page without refreshing the entire page is one characteristic of AJAX. AJAX also comes with a JavaScript-based HTTP request/response framework that can be integrated into a web page.

4 Document Object Model (Dom)

Every JavaScript-enabled web page is effectively represented by a browser as a W3C Document Object Model instance (DOM). The Document Object Model (DOM) is a platform-agnostic and language-neutral object model for describing XML-based documents that enables programs and scripts to interactively access and edit the document's content, structure, and style.

5 Extensible Markup Language

The XML (eXtensible Markup Language) definition and standard are used to create self-descriptive markup languages. It's widely used in data transformation and integration frameworks to make structured data transmission and integration easier. XML is utilized as a standard for document annotation and serialization format for the data in many web-enabled contexts.

6 Javascript Object Notation

JavaScript Object Notation (JSON) is a basic data-exchange format built from JavaScript object literals. It is a string-based format. JSON is simple to read and write for users and to parse for JavaScript engines. String literals in a JSON object can represent strings, integers, Booleans, arrays, and objects.

6.3 Technique for Enterprise Mashup Development

--Step 1, Identify the issue domain and describe Business goals and Performance measures.

--Step 2, Determine the IT infrastructure, including all program semantics, assets, relevant information and sinks accessible in that domain and web enabled.

--Step 3, Identify technical needs, including a catalogue of all external interfaces that should be used (data, services, and basic information) and the definition of new resources, services, and information connected to those services. Additionally, define new processes, as well as the services and information that are associated with them.

--Step 4, Determine the technical set, focusing on choosing your technology stack and implement, for example, utilizing SOA technologies and test/evaluate.

--Step 5, Keep your mashup up to date by Creating a version control system and data integrity system.

TABLE 1
MASHUP DEVELOPMENT TECHNIQUES

Category	Steps of Mashup development techniques
Techniques of on fly service	- Component Retrieval - Composition of a programming interface

	- Construction of the user interface
Techniques of Presentation based mashups	- Combining data for presentation - Using the EXML HTTP request object and AJAX - Document object model - Extensible markup language
Techniques of Enterprise mashup development	- Identify issue domain - Determine IT infrastructure - Identify technical needs - Determine technical set - keep mashup up to date

7 MASHUP DEVELOPMENT CHALLANGES

This section has a detailed discussion on the challenges that are faced while adopting the mashup technology by enterprises and users along with challenges in development of mashup tools by mashup developers.

7.1 Detection of Event's Performance

The detection event's performance is limited by the usage of only one social media site as a source of information. On the one hand, this results in a lack of details that explain the events. On the other hand, the usage of a single social media platform limits the diversity of event-related data sources.

7.2 Combination of Data

Many mashups seek public user feedback as part of their IT application design. This is a double-edged sword, as illustrated by the wiki application domain: it may be fairly effective since it allows for open contribution and best information development, but it can also be vulnerable to inconsistent, faulty, or purposefully misleading data entry. The latter can raise questions about the reliability of data, lowering the mashup's value.

7.3 Integrating Data Sources with Inconsistency in Semantics

Developers of mashups have challenges in determining common semantic meaning amongst multi-source information. They may realize the data they intend to integrate is not acceptable, necessitating more processing, despite the missing or incomplete data mappings. The customer opportunity record, for example, may be inputted inconsistently, with common abbreviations for names like "CustomerOpp" in one CRM system and "Cus Opp", making automated reasoning about equality problematic, even with strong standards.

7.4 Lack of Micro Formats

The development of parsing and collection tools, as well as data models, necessitates a large amount of reverse engineering. Even in the best-case scenario, a reorganization of how the source site shows its information (or shuttering and abandonment) is all it takes to interrupt the integration process and cause mashup application failure. Micro format is an emerging technology that allows both humans and robots to understand web pages. However, there are now just a handful micro format requirements, and most websites do not use them. If websites adhere to semantic standards while processing data on the web, screen scraping will become easier.

7.5 Lack of Adaptivity

Another important challenges that is being faced by practitioners in manual or automatic creation of the mashups is the lack of adaptivity. It means that if the sources of data are changed or altered by the provider change their structure or behavior the mashup application need to be restructured by the end users

7.6 Technical Expertise Requirement

Manual development requires skills and expertise in web technologies to create complex mashups. But it is difficult for casual user, power user to create personalized mashup application in appropriate time, so a relatively small set of developers is not able to create mashups that serve the changing and very different needs of a huge mass of end-users.

7.7 Mashing Data Togather

In In-Process Data Mashup, it might be difficult to mash data together in a web browser. Browser-based scripts and technologies may or may not be suitable for data integration depending on the kind of data.

7.8 Data Accessability

Trying to access services located on a site different than the one from which the initial page was downloaded is one of the most challenging difficulties with presentation-oriented mashups. Most mainstream web browsers use a sandbox security paradigm, which prevents a web page from accessing a site or service that is not hosted on the same host as the page.

7.9 Security Issues Related to Dynamic Communation Paradigm

To a web host, a JavaScript container (browser). AJAX allows web pages to behave more like desktop apps, such as requiring less page refreshing and using dynamic data updates. This dynamic communication paradigm, on the other hand, opens the door to harmful programs.

7.10 Efficiency-Related Challenges

Mashup development also has challenges related to efficiency. Some of them are given bellow.

1 Categorization of Mashups

Some Web sites, such as programmableweb.com, already catalogue mashups and provide a search mechanism for finding them. Mashup makers can add their mashups to the list and share them with the rest of the world. What's needed, though, is a directory that categorizes and saves the mashups in a standard manner.

2 Data Integrity

Using mashups is a convenient method to construct novel applications, but it can be a source of data integrity issues if end-user changes do not match the fundamental commitment. Additional worry is that integrity issues may arise if, for example, an end-user discovers a resource that adds value to the data or feature used in the mashup. Then mashups must be tweaked in real time. As a result, when operating a mashup, control measures that maintain the mashup's integrity against end-user modifications should be addressed.

3 Web Enabling Data

Many data and features are now not available via RSS, HTML, or Online services since they are not established up on the Web. To make larger resources "Web-enabled," standards and technologies that make accessing and connecting resources to the Web easier are needed. Furthermore, since the content is capsulated with the presentation layer, some data available on the Web cannot be utilized for "mashing." As a result, methods that facilitate the production of data mashups, as well as tools that give functionality to detach data from numerous sources from their representation, are required.

4 Security & Uniqueness

If the mashup comprises sensitive data or necessitates security log-ins to submit certain data, security issues come up. This demands systems to regulate the user interaction and data protection.

5 Sharing & Reuseability

There should be a mechanism for sharing mashups with other to help them reusing. A legal mashup needs to be in a design that is understandable by tools and measurable. The challenges that must be fulfilled in this context include immediate access to mashups, fast mashup search capabilities, and minimalist structures that enable seamless mashup reuse even for non-coders.

6 Authorization Certification

A license that certifies the mashup can be issued by the proprietor of such a metadata repository. Because there are currently no certification processes in place to ensure the mashup's reliability to end users.

7 Version Management Procedures

A version control method is necessary to keep the mashup material up to date, which automatically notifies the mashup owner about revisions to the integrated runtime environment (assume the mashup is built upon numerous APIs).

7.11 Manual Construction

Manual mashup development (coding or script writing) necessitates knowledge of web technologies and coding in order to produce complicated mashups.

7.12 Time Consumption

It's challenging for both casual and power users to construct customized mashup apps in a timely manner.

7.13 Background Knowledge Requirement

Selecting data manually requires not only knowledge of the data sources' locations (e.g. URLs), but it also requires to have information about the organization and semantics of the information that can be retrieved, because otherwise the human agent would be unable to determine whether a particular data source is in accordance with the problem at hand. In fact, it's possible that the end-user is unaware of many of the available sources and is therefore restricted to a small number of well-known ones, which may or may not meet his needs for credibility, accuracy, privacy, and other factors in a given circumstance, negatively impacting decision quality. Therefore, shift to semantics is the need but still some of the knowledge is required.

7.14 Wire Based Approach Challenges

1) Gives Depression to Users

Users are under pressure to choose suitable wire-oriented architecture building blocks because they must transform an issue's problem and domain-specific, high-level depiction into a conceptual, precise, and technical flow of modules, blocks, and other elements.

2) Adaptation Lack

This implies that if the provider's data sources and functions alter their structure or behavior, the end users must completely redesign the mashup program. Based on the findings, we may conclude that the production of mashups by manual or semi-automatically has certain inherent limits.

7.15 Consumer Vs Enterprise Approach Struggle

Business mashup application models attempt to predict what will work best in beforehand, and they may be overlooking the massive Web environment that is already showing extremely successful models for mashup pieces and integration tactics on a broad scale. A skill and an open/closed barrier are also included. The consumer web model flourishes in an open source setting because it benefits from a huge pool of talent. Their main focus is on simplicity of use, with minimal attention paid to problems such as security, installation challenges, sustainability, and so on. The business model, on the other hand, prefers a restricted environment to protect its corporate interests, which results in a limited talent pool. Additionally, this model must invest extensively in major problems like as security, installation, updates, and maintenance.

7.16 Widget Format Standardization

To construct a graphical tool that can completely support mashup creation and achieve the productive capacity, a uniform widget format is necessary. Organizations aren't yet ready to start sorting out which widget models are right for them, and there's no apparent answer on the road. Both Microsoft and Google have their own widget models, but NetVibes' Universal Widget Architecture is particularly attractive (UWA).

7.17 Shortage of Web Service for Enterprise

It will take years and years to build a full SOA-based framework. In the meanwhile, if we are to fully utilize the potential of mashups, we must swiftly service-enable our barriers of web-based content. Fortunately, firms like Kapow, Intel, and Yahoo Pipes are already proposing solutions to help reduce this problem. It uses graphical tools to convert Web 1.0 material into a Web 2.0 friendly service style like RSS in a matter of minutes. As a result, widgets are perfect for intake since they provide a simple service.

7.18 Issues in Enterprise

To get a presence in the enterprise, mashup technologies must be able to provide compelling insights about single sign-on (SSO), LDAP, JSR168 (portals/portlets), historical integration, administration, measurement, RSS policy, accountability, and other concerns. Most mashup tools still fall short in these areas, and they are unlikely to gain general acceptance unless

these concerns are resolved. [33]

7.1 Creating Mashup Settings

This is a big challenge of mashup technologies that they are designed in such a manner that novice and people with less knowledge of technology may construct mashup applications in a variety of areas with ease and effectiveness. One of the most challenging difficulties is creating mashup settings that are suitable to end user development for non-technical users. [32]

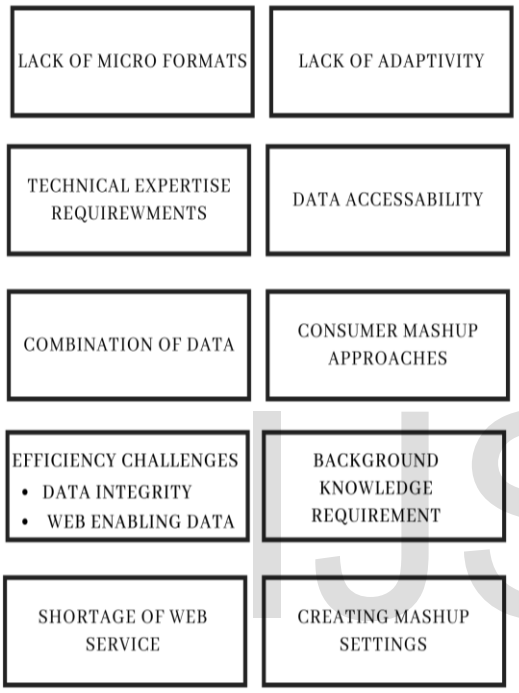

Fig. 5. Challenges of mashup development being mashup creators and mashup technology users.

8 Findings & Discussion

For this study, we have conducted a literature review for having knowledge of various types of approaches that are used for mashup development and techniques that are used for them. We have also elaborated the challenges associated with the approaches that are being proposed. Summary of the classification in which mashup approaches can be classified is elaborated in Table 1.

TABLE 2
Classification of Mashup Development & Approaches

Category	Sub Categories	Examples	Description
Media Mashup Approaches		a map featuring top-ranked high school features directly on the map.	Media mashup approaches are those that are used for recombination of digital media files that contain any or all of the following elements: texts, images, audios, videos, and animations.
Web Mashup Approaches		special mashup tools like IBM's mashup tools, or dashboard systems like Netvibes	Web mashups are web applications that are created by combining and repurposing data and functions that are available on the Internet.
Process mashup approaches	Manual, Semi Auto (Spreadsheet Oriented, Widget Based, Demo Based), Auto		Approaches that are used for the development of process oriented mashups including auto, manual and semi auto approaches.
Mashup Building Environments Approaches	Consumer Mashups, Enterprise Mashups	Dapper, Grazr, DERI pipes, Apatar, Data mashups, IBM InfoSphere MahsupHub	Approaches used for building mashup like on both enterprise and consumer level.
Mashup Modality Approaches	Extraction, Flow mashup Approaches	Apatar, Data mashups, DERI pipes, Intel mash maker	Approaches used for building mashups that involve styling and modality of the mashups.
Mashup asset Integration Approaches	Functionality, Presentation, Mapping, Data mashup Approaches		Approaches that are used for building mashup that involve integration of different types of mashups.
Service based Approaches			The approaches that use method for transforming old systems into service-oriented mashups
Passive Approaches			It focuses on creating plugins that operate with the user's existing browser, watch what they're looking at, and offer relevant sites for possible mashing.
Proactive Ap-			It allows users to create mashups in complex

IJSER

proaches	integration domains by first providing samples of what the end mashup should look like

TABLE 3
MASHUP DEVELOPMENT CHALLANGES

Category	Challenges
Enterprise mashup techniques and approaches	- Shortage of web service for enterprise - Issues in enterprise - Creating mashup settings
Data mashups	- Data integrity - Data accessibility - Integrating data sources with inconsistency in semantics - Adaption lack - Lack of micro formats - Combination of data - Mashing data together
Challenges faced by users	- Background knowledge requirements - Wire based approach gives depression to the users. - Time consumption - Security challenges - Categorization of mashups
Challenges of wire based approach	- Time consumption - It gives depression to users - Manual Construction

9 CONCLUSION

As we know the trending technology of mashups in field of End-user Development from which enterprise and individuals are getting benefitted to make their processes fast. Researchers are currently working on different aspects of mashup development including the approaches that could help in development of mashup development and also the challenges that are associated with those approaches and techniques being proposed. While adoption of mashup development, users find it difficult to gather information from different sources of information that is appropriate information that applicable to the requirements of the end user enterprises and individuals. In this study, we have studied the literature and provided a literature review of the proposed approaches and techniques for the mashup development. Moreover, we have classified the mashup development approaches into different categories depending of the nature of their use. We have also demonstrated the challenges that are faced at different levels of mashup development and adoption by users and mashup creators for different types of mashups.

REFERENCES

[1] Hartmann, Björn, Leslie Wu, Kevin Collins, and Scott R. Klemmer. "Programming by a sample: rapidly creating web applications with d. mix." In Proceedings of the 20th annual ACM symposium on User interface software and technology, pp. 241-250. 2007.

[2] Lars Grammel and Margaret-Anne Storey. 2010. A survey of mashup development environments. The smart internet: current research and future applications. Springer-Verlag, Berlin, Heidelberg, 137–151.

[3] Kulathuramaiyer, Narayanan. (2007). Mashups: Emerging Application Development Paradigm for a Digital Journal. J. UCS. 13. 531-542.

[4] Yu, Jin & Benatallah, Boualem & Casati, Fabio & Daniel, Florian. (2008). Understanding Mashup Development. Internet Computing, IEEE. 12. 44-52. 10.1109/MIC.2008.114.

[5] Rabia, Mulla & Shukur, Zarina & Wills, Christopher & Latih, Rodziah & Patel, Ahmed & Liu, Na. (2010). A Study of Mashup as a Software Application Development Technique with Examples from an End-User Programming Perspective. Journal of Computer Science. 6. 10.3844/jcssp.2010.1406.1415.

[6] Hummer, W., Leitner, P. and Dustdar, S., 2010, December. A step-by-step debugging technique to facilitate mashup development and maintenance. In Proceedings of the 3rd and 4th International Workshop on Web APIs and Services Mashups (pp. 1-8).

[7] Ravallion, Martin. "Mashup Indices of Development." The World Bank Research Observer, vol. 27, no. 1, Oxford University Press, 2012, pp. 1–32, http://www.jstor.org/stable/23262819.

[8] EDBERG, DANA, et al. "Methodology Mashups: An Exploration of Processes Used to Maintain Software." Journal of Management Information Systems, vol. 28, no. 4, M.E. Sharpe, Inc., 2012, pp. 271–303, http://www.jstor.org/stable/41713864.

[9] Cappiello, Cinzia, Maristella Matera, Matteo Picozzi, Florian Daniel, and Adrian Fernandez. "Quality-aware mashup composition: issues, techniques and tools." In 2012 Eighth International Conference on the Quality of Information and Communications Technology, pp. 10-19. IEEE, 2012.

[10] Scotch, Matthew, Kevin Y. Yip, and Kei-Hoi Cheung. "Development of grid-like applications for public health using Web 2.0 mashup techniques." Journal of the American Medical Informatics Association 15, no. 6 (2008): 783-786.

[11] Minhas, Sumaira S., Pedro Sampaio, and Nikolay Mehandjiev. "A framework for the evaluation of mashup tools." In 2012 IEEE Ninth International Conference on Services Computing, pp. 431-438. IEEE, 2012.

[12] X. Liu, Y. Hui, W. Sun and H. Liang, "Towards Service Composition Based on Mashup," 2007 IEEE Congress on Services (Services 2007), 2007, pp. 332-339, doi: 10.1109/SERVICES.2007.67.

[13] XU, H.Y., SONG, M.N., Hui, C.H.E.N. and SONG, J.D., 2008. Research on SOA based mobile mashup platform for telecom networks. The Journal of China Universities of Posts and Telecommunications, 15, pp.31-36.

[14] Bo Jiang, Pengxiang Liu, Ye Wang, Yezhi Chen, "HyOASAM: A Hybrid Open API Selection Approach for Mashup Development", Mathematical Problems in Engineering, vol. 2020, Article ID 4984375, 16 pages, 2020. https://doi.org/10.1155/2020/4984375

[15] Fischer, Thomas & Bakalov, Fedor & Nauerz, Andreas. (2009). An Overview of Current Approaches to Mashup Generation.

[16] C. Li, R. Zhang, J. Huai and H. Sun, "A Novel Approach for API Recommendation in Mashup Development," 2014 IEEE International Conference on Web Services, 2014, pp. 289-296, doi: 10.1109/ICWS.2014.50.

[17] Anarfi, Richard, and Kenneth K. Fletcher. "A reinforcement learning approach to web api recommendation for mashup development." In

2019 IEEE World Congress on Services (SERVICES), vol. 2642, pp. 372-373. IEEE, 2019.

[18] Gu, Qi, Jian Cao, and Qianyang Peng. "Service package recommendation for mashup creation via mashup textual description mining." In 2016 IEEE International Conference on Web Services (ICWS), pp. 452-459. IEEE, 2016.

[19] Q. Zhao, G. Huang, J. Huang, X. Liu and H. Mei, "A Web-Based Mashup Environment for On-the-Fly Service Composition," 2008 IEEE International Symposium on Service-Oriented System Engineering, 2008, pp. 32-37, doi: 10.1109/SOSE.2008.9.

[20] Albinola, Matteo & Baresi, Luciano & Carcano, Matteo & Guinea, Sam. (2009). Mashlight: a Lightweight Mashup Framework for Everyone. Proceedings of the 18th international conference on World Wide Web, WWW'09.

[21] Aghaee, Saeed, and Cesare Pautasso. "End-user programming for web mashups." In International Conference on Web Engineering, pp. 347-351. Springer, Berlin, Heidelberg, 2011.

[22] Cheung, Kei-Hoi, Vipul Kashyap, Joanne S. Luciano, Huajun Chen, Yimin Wang, and Susie Stephens. "Semantic mashup of biomedical data." Journal of biomedical informatics 41, no. 5 (2008): 683.

[23] J. Jeffrey Hanson, author of Mashups: Strategies for the Modern Enterprise, points out some of the most widely used styles, techniques, and technologies to build mashups.

[24] Paredes-Valverde, Mario & Alor-Hernández, Giner & González, Alejandro & Valencia-García, Rafael & Jiménez-Domingo, Enrique. (2013). A systematic review of tools, languages, and methodologies for mashup development. Software: Practice and Experience. 45. 10.1002/spe.2233.

[25] Troudi A., Zayani C.A., Jamoussi S., Amous I. (2017) A New Social Media Mashup Approach. In: Madureira A., Abraham A., Gamboa D., Novais P. (eds) Intelligent Systems Design and Applications. ISDA 2016. Advances in Intelligent Systems and Computing, vol 557. Springer, Cham. https://doi.org/10.1007/978-3-319-53480-0_67

[26] Majer, Frederic, Martin Nussbaumer, and Patrick Freudenstein. "Operational challenges and solutions for mashups–an experience report." In 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), held in conjunction with 18th International World Wide Web Conference (WWW 2009). 2009.

[27] Gnanasekar, jm. (2011). Towards Development of QoS Based Data and Service Oriented Mashup Platform. CiiT International Journal of Networking and Communication Engineering. volume 3. 312- 0.

[28] A. F. M. Huang, S. B. Huang, E. Y. F. Lee and S. J. H. Yang, "Improving End-User Programming with Situational Mashups in Web 2.0 Environment," 2008 IEEE International Symposium on Service-Oriented System Engineering, 2008, pp. 62-67, doi: 10.1109/SOSE.2008.56.

[29] Koschmider, Agnes & Torres, Victoria & Pelechano, Vicente. (2009). Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups.

[30] Goussis, Lampros & Foukarakis, Ioannis & Kallergis, Dimitrios. (2010). Aspects and Challenges of Mashup Creator Design. 10.1109/PCI.2010.22.

[31] Daniel, Florian, Maristella Matera, and Michael Weiss. "Next in mashup development: User-created apps on the web." IT Professional 13, no. 5 (2011): 22-29.

[32] Brandon, Beemer & Gregg, Dawn. (2009). Mashups: A Literature Review and Classification Framework. Future Internet. 1. 10.3390/fi1010059.

[33] Peenikal, Sunilkumar. "Mashups and the enterprise." Strategic white paper (2009).

IJSER