Web of Goals: A Proposal for a New Highly Smart Web

Meriem Benhaddi
Faculty of Sciences and Techniques, Cadi Ayyad University, BP 549, Marrakesh, Morocco

Keywords: Web Future Generation, Smart Web, Smart Life, Web 4.0, Web 3.0, Web 2.0, Goal, End User, Interaction, Context, Adaptation, Mashup, Composition, Artificial Intelligence.

Abstract: Since Web use revolution known as Web 2.0, and the birth of a third version which is the semantic Web or Web 3.0, users needs have kept changing and becoming more demanding in all aspects of life (health, education, economy, etc), giving rise to a new wave of principles that have emerged to constitute a new smart web called Web 4.0, encompassing new principles, concepts and technologies that bring new solutions. Until today there is no exact definition of Web 4.0, much less a definition of architectural principles; however, Web 4.0 consists of the new Web generation that is built on Web 3.0 and Web 2.0 principles, in addition to new notions such as artificial intelligence, mind controlled interfaces and intelligent goal searching engines. Web 4.0 offers more autonomy and creative opportunities to end users in order to quickly reach their goals by efficiently express their needs, create new applications or adapt existing ones to their personal contexts. In this paper, we give our own definition to the new smart Web 4.0 by highlighting what makes it different from the earlier Web versions; then we propose architecture elements that will allow transforming the Web into an Ultra-Intelligent Electronic Agent. We introduce a motivational scenario that illustrates and nurtures the feasibility of our point of view.

1 INTRODUCTION

With the recent economic and technological changes that the world knew in the recent decades, the principles governing the Internet are also subject to big changes due to new usages, new needs and new trends. The new versions of the Web - Web 2.0 and Web 3.0 – (O'Reilly, 2007) reflect the need for end users - with no technical knowledge and who interact with information systems solely as final information consumers in the context of daily life or daily work with no intention to produce other systems (Cypher, 1993) - to face new requirements of speed, lightness and mobility.

Web 2.0 is a term that was first coined in 2004 by Tim O'Reilly, that means all the technologies and uses in the World Wide Web that followed the initial version of the web; In particular, Web 2.0 is about new interfaces allowing end users with little technical knowledge to interact in a simple way with both the content and structure of pages but also with other end users, thus creating the social Web that encompasses collaboration and sharing, publishing in blogs and micro-blogs, social networking, aggregation and composition, feedback and viral marketing, Co-development, Crowdsourcing, Crowdfunding and SaaS (Software as a Service).

On the other side, Web 3.0 is an extension of Web 2.0 and designates the semantic web. The vision of the semantic web is "turning the World Wide Web to an environment in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee et al., 2001). The semantic theory provides an "account of "meaning" in which the logical connection of terms establishes interoperability between systems" (Shadbolt et al., 2006).

Fortunately, as the needs of web users keep changing and becoming more demanding in all aspects of life, the Web accompanies these needs and invents new principles, concepts and technologies that bring new solutions. In fact, in recent years, new challenges emerged and consisted for computers to help users meet their goals (needs) that range from recording sales to a complex use case involving various services (software, platform or infrastructure services). Computers have to capture users' requirements and intelligently transform them into adequate services. In other words, the Web has to be parallel of and as powerful as the human brain (Choudhury, 2014; Aghaei et al.,

Benhaddi, M.
Web of Goals: A Proposal for a New Highly Smart Web.
DOI: 10.5220/0006250306870694
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 687-694
ISBN: 978-989-758-248-6

2012). This new shift in the Web led to what is known today by the Web 4.0 or particularly the WebOS (Web Operating System), which is playing the role of a middleware that will start functioning like an operating system to become the Ultra-Intelligent Electronic Agent.

To the best of our knowledge, until today there is no exact definition of Web 4.0; however there are some characteristics that allow differentiating it from the earlier versions. These characteristics consist mainly in moving towards using artificial intelligent techniques to be as a massive web of highly intelligent interactions (Choudhury, 2014). Web 4.0 applications are mind controlled interfaces, intelligent goal searching engines, intelligent agents, mobile technologies, cloud computing and services, etc (Nedeve and Dineva, 2012).

Our Web 4.0 Definition: We define the Web 4.0 as the new Web of goals as its mission is to intelligently detect end users intentions and goals and propose them solutions. In fact, while the semantic web or Web 3.0 links semantically the web resources with a descriptive layer, the new Web 4.0 will create a new layer that offers goal oriented links and a set of intelligent operators that connect and transform resources from the functional point of view (see figure 1). This new smart Web will allow us using smart objects in a smart way in order to lead a smart life in the fields of health, education, business, administration, leisure, etc.

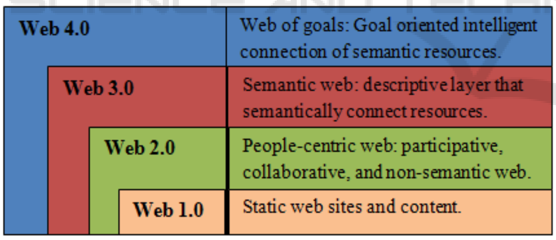

Figure 1: World Wide Web evolution: from Web 1.0 to Web 4.0.

In order to make industries aware of this new trend and make it easy to create Web 4.0 frameworks, we make in this paper an architecture proposal that formalize this new version of web and offer new ideas and principles that will help building Web 4.0 technologies. Our architecture is dedicated to describe the functionalities that should be handled and implemented by a Web 4.0 frameworks, and is not a technological architecture.

The remainder of this paper is organised as follows. Section 2 presents some motivational scenarios which triggered the design of our architecture. In Section 3, we review the state of the art. Section 4 introduces the proposed architecture and presents its major building blocks, and finally section 5 concludes our paper and gives directions for future work.

2 MOTIVATIONAL SCENARIO

How many of us have wished to have an intelligent platform that understands our goals and intentions, offers us the adequate applications that meet our needs, allows us to easily create new applications from small ones and automatically adapts to the changes that may occur in our context? Such a platform would increase end users satisfaction, richness of the web and productivity within enterprises.

To illustrate our point of view, let us introduce some motivational scenarios that nurture the feasibility of our approach: Our end user, Alice, wants to pick up her friend in the airport. In her car, Alice uses the city navigation service that helps her to get to the airport before the arriving time of her friend's plane. After arrival at the airport, Alice wants to use the airport navigation service in order to get different information: the gate number of her friend's flight, a closer parking, a currency exchange office and a restaurant closer to the gate in order to make a reservation 30 minutes after the arriving time of her friend. Mark is a second user that has the same goal as Alice but is situated in a different airport. Mark wants to use the application created and stored previously by Alice after adapting it to his new context, by preserving the purpose of the application and changing the used services by other ones meeting the new context and proposed by the intelligent platform.

3 RELATED WORK

Users have goals (needs) and want computer systems to help meet them. The first primordial thing that computer systems have to do is to capture these users' needs − there are several ways to capture tasks, goals and system requirements −, then the system has to find or to create a service that meets this need by composing existing services. This section discusses existing techniques for capturing users' needs and composing services.

3.1 Task and Goal Modeling

According to Alberto and Troutman (2003), task analysis is the process of breaking down a complex task into its smaller steps or components. Paterno (2001) states that task models are logical descriptions of the activities, manual or mental, to be performed in reaching user's goals. A goal is being defined as a prescriptive statement of intent about some system (existing or to-be) (Bernardo and Inverardi, 2003). Task modeling has been used for different purposes and are particularly useful when designing and developing interactive systems (Kritikos, Plexousakis and Paterno, 2014) (Lankhorst, 2012).

According to Paterno (2003), task models can be presented syntactically (textually or graphically) (Unified Modelling Language (UML) (Miles and Hamilton, 2006), ConcurTaskTrees), with a set of operators for task composition, or with a level of formality. Lankhorst (2012) list a number of task models that have been proposed in the literature: Cognitive Task Analysis, ConcurTaskTrees, Hierarchical Task Analysis and KAOS.

3.2 Web User Goal Modeling

In the field of modeling web searching user goals, Gawade and Chhajed (2014) proposed an approach that captures end user web query goals based on their feedback sessions; the authors allow predicting user's intention on web searching in order to improve search engine relevance and user knowledge. While Liu, Liberman and Selker (2002) proposed an adaptive search engine interface that uses natural language processing to parse a user's search goal, and uses "com-mon sense" reasoning to translate this goal into an effective query, Rose and Levinson (2004) proposed a framework for search goal that allows storing web search goals as a flat list that classifies user queries.

In order to improve computer system development and facilitate designing and adding new computer features, Nielsen (1994) describes a technique for extending a task analysis based on goal composition principle that was suggested by Clayton (1990). This technique helps anticipating future users' needs and extending and combining basic needs with general meta-goals, called goal composition mechanisms. Nielsen (1994) discussed three main categories of goal composition mechanisms which are: generalization mechanisms, integration mechanisms and user control mechanisms. In the other hand, Bolchini and Mylopoulos (2003) used task analysis and extended it to propose a goal-oriented application design approach; the authors used a goal decomposition based on OR and AND relationship operators.

We have not found an existent work that joins at the same time task modeling, end user goals and service creation. The next section presents and discusses the end user service composition known as Mashup paradigm which emerged in the Web 2.0.

3.3 End User Service Composition with Mashup

Mashup emerged as a new paradigm of the Web 2.0 (O'Reilly, 2007) that enables the user generation of services and allows end users without technical knowledge to personalize and customize their applications (Liu, Huang, Zhao, Mei and Blake, 2014). Applications created with Mashup are called situational applications because they allow quick solving of problems that occur frequently in specific situations (Ma, Lu, Liu, Wang and Blake, 2013). The Mashup or end user service creation can be a great help for goal creation; in fact, each goal being concretely reached by one or more services, the end user service creation, enhanced with semantic, patterns and logic, allows easy and fast end user goal achievement.

The Mashup architecture helps to fill in the gap between the end-user and the software development; however, no global overview is offered about the composed services, their relationships, their semantics and the operators that could be applied on them. In fact, the composed services are stored as rigid packages and thus are not considered as goals. Our proposal offers a global perspective on end user goals: web user goal modeling, goal storing, goal creating with composition process, goal transforming, etc. In the next section, we present our way of seeing and defining goals and our new Web 4.0 architecture.

4 PROPOSED WEB 4.0 ARCHITECTURE

4.1 Goal Mashup

A goal is either a desired modification of state or an inquiry to obtain information on the current state. By referencing our motivational scenario in section 2, an example of a goal, named "Airport services", can be decomposed into multiple sub-goals which are: getting to the airport, making a restaurant reservation, locating a currency exchange and paying the parking. The creation of this goal will be realized by composing and combining all the above sub-goal. When involving end users that do not have advanced skills in information technologies in the process of goal creation, we should take into consideration their mental model that characterizes their level of perception of the used system (Mehandijev, Lecue, Wajid, and Namoun, 2010). This mental model differs from the mental model of programmers; in fact, when creating new applications, end users try to achieve a new goal by composing existing sub-goals ; each sub-goal being represented by a service. End users are not interested in the composition process per se, but in getting and achieving the desired results; thus, we define our approach as goal or task oriented instead of service oriented in order to respect the mental model of end users.

The goal composition process leads to an application that is composed of a set of goals; similarly to the end user service composition-applications called service mashups (Benhaddi, Baïna and Abdelwahed, 2012), we call the applications resulting from the end user goal composition process: *goal mashups*.

4.2 Goal Patterns

When compositing goals/services in response to a new need, the inexperienced end user faces many challenges (ex. determine the types of resources, find resources that meet the end user criteria (quality, price, etc.), determine necessary actions for the use of interfaces (selection problems), determine how to arrange and coordinate resources (integration), etc). The system has the role of helping end users to answer these different questions, by suggesting goals, resources, providing guidelines for the coordination of goals and providing feedback and documentation for each selected action. To achieve this, we provide the end user with a set of goal prototypes or goals patterns, that we define as common and repetitive use cases or goal mashups, which can also be called end users experience patterns since they are driven from the end users experience. They provide answers to questions like "How to automate the execution of two consecutive tasks - eg. Use the city navigation service and then switch to the airport navigation service - in response to a triggered event? - ex. presence of a person in an airport. While software design patterns are derived from the experience of the software developers, goals patterns are created, improved and enriched by end users themselves. The database of goals patterns being built through the experience of end users, the system will score the various components, depending on the frequency of use, and thus offer to the end user the best one - which has the highest score.

4.3 Web 4.0 Architecture

In this section, we present our proposal for a new Web 4.0 architecture. The layers of this architecture describe the functionalities that should be handled and implemented by a Web 4.0 framework, and do not propose the enabling technologies of each layer. The enabling technologies will be presented as a future work.

The Web 4.0 architecture presented in figure 2 includes ten vertical layers that stand the process of creating a goal-Mashup application.

4.3.1 Goal Description Language or Goal Description Model

The aim of this first layer is the representation of goals or tasks. Goals need to be formalized using a description model or language that should be expressive by offering rich operators. Presented in section 3.1 of this paper, task models, such as UML and ConcurTaskTrees, are to be used to describe end user goals. A set of operators for task composition is also a good candidate for this task; we will define a set of operators for our composition language that will be used in our Web 4.0 architecture (see section 4.3.4).

This layer offers a syntactic representation, which is the lowest level of abstraction where the goal is represented as an atomic concrete service or a combination of concrete services. This representation is the realization of the semantic representation. A semantic representation can be realized by one or several syntactic representation.

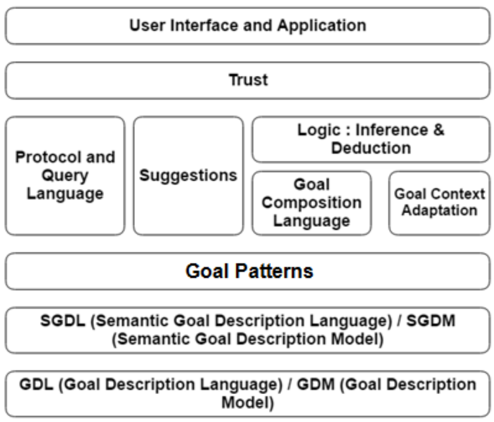

Figure 2: Web 4.0 architecture overview.

4.3.2 Semantic Goal Description Language or Semantic Goal Description Model

This layer offers a semantic representation of goals, which is the highest level of abstraction where the application is represented as a composition of goals/sub-goals. This representation describes the end user request; in fact, the end user is interested in achieving an objective whatever the used services are. When the end user context changes (see section 4.3.5), this semantic representation does not change; in fact, the end user goal/request stays the same while its realization can change.

A goal is also described by several semantic information as its type, its physical environment of execution, its objective, its frequency of execution, its duration of execution, its degree of importance and end users profiles that can execute this goal. This information can be extended and enriched in order to create semantic relationships between goals, which is the role of the goal patterns layer (see section 4.3.3).

4.3.3 Goal Patterns

The goal patterns layer represents the relationships between goals based on goals descriptive information (type, physical environment of execution, objective, frequency of execution, duration of execution, degree of importance and end users profiles that can execute the goal). Classificatory schemas are to be created and stored in order that intelligent agents can interpret data and make inferences (see section 4.3.6). As previously pointed out (section 4.2), goal patterns are created, improved and enriched by end users themselves.

4.3.4 Goal Composition Language

This layer offers a language for creating new goals by composing existing ones. The composition language should allow the composition of simple and complex scenarios and use cases; rich entities, operators and integration patterns are required to build such advanced integration scenarios.

In our architecture, the goal composition language model is based on the Enterprise Integration Patterns (EIPs) (Hohpe and Woolf, 2003) that are powerful for allowing the realization of complex integration scenarios. Enterprise Integration Patterns propose the best and common solutions to integration problems. Therefore, when EIPs are used, they enhance the quality of the integrated applications. EIPs consist of six groups of patterns: messaging channels, message construction, message routing, message transformation, messaging endpoints (message is the data unit that transfers from one goal to another) and system management. The syntactic and semantic formalization of the goal composition language is presented by Benhaddi, Baïna and Abdelwahed in (2013). Table 1 lists the operators that are required for an efficient and rich goal composition.

Table 1: Goal composition operators.

Group of operators	Role of operators	Examples of operators
Links between goals	Allow linking goals, ordering and arranging them.	Simple link (sequence) Broadcast link Adapter link
Goal data construction	Specify the data manipulated by goals: their inputs and outputs.	Command data Document data Event data Request-reply data
Goal data transformation	Allow transforming the content or the format of data in order to prepare it to be manipulated by goals.	Data enricher Data filter Data enveloper
Goal routing	Specify the types of links between goals to forward data to a specific destination.	Data-based router Goal filter Goal aggregator Goal splitter
System management	Dealing with performance, exception, bottlenecks, etc.	Validator and tester of goals data Links purger

4.3.5 Goal Context Adaptation

With the pervasive use of mobile devices and the need for ubiquitous computing, the issue of "context" is becoming a hot topic in human computer interaction research and development (Anind, 2001). For an efficient user-centric goal creation, the "context" dimension is needed as a new requirement in order to enhance ease of use, efficiency and consequently end users satisfaction. In fact, with a semi-automatic goal composition where the end user performs a goal composition in response to his need (request), the system should be able to automatically adapt the composite goal if any change occurs in the end user context (change in place or/and in time or/and in the device, event occurrence, etc).

Once the new application is created by end users, it has two representations in two different levels:

- Semantic representation: this is the highest level of abstraction where the application is represented as a composition of goals/sub-goals. This representation describes the end user request whatever the used services are. When the end user context change, this semantic representation stays the same while its realization can change.
- Syntactic representation: this is the lowest level of abstraction where the application is represented as a composition of concrete services. This representation is the realization of the semantic representation (one goal can have one or several syntactic representation).

In order to automatically adapt the composite application to any change in the end user context, the system has to search for new services that realize the end user goal and that correspond to the new context of the end user. The new goal could be retrieved from the goal patterns (see section 4.3.3) using the operators offered by the "Logic: inference and deduction" layer (next section).

4.3.6 Logic: Inference and Deduction

The major role of this layer is to make inferences and deductions on stored goals. In fact, when the user formulates a new need using the query language (see section 4.3.8), the inference engine tries to offer to the end user a goal that meet all the expressed requirements. For this purpose, the end user request is divided into two parts on which the logic engine operates separately:

1) Functional dimension: represents the functional needs retrieved from the end user request.
2) Contextual dimension: represents the contextual needs retrieved from the end user request.

When analyzing the end user request, the logic engine could face one of the following three situations:

- A stored goal meets accurately and exactly (functionally and contextually) the end user request, which will be offered to the end user.
- A stored goal meets the end user functional needs but in a different context; in this case, the goal is transmitted to the Context Adaptation engine (see section 4.3.5) in order to adapt the old goal to the new end user context.
- The expressed functional needs are not offered by any stored goal; in this case, the end user request is divided into multiple sub-requests; each sub-request will undergo the same process as the initial request, and the sub-goals – results of this operation – are transmitted to the composition engine (see section 4.3.4) in order to be assembled to create a new goal that meet the end user request.

Based on the work of Rui and Butler (2003), we define the basic logic operators – the list could be extended:

Table 2: Goal logic operators.

Operator	Role of the operator
Inclusion	One goal is containing another goal which is a sub-goal means that the latter is a behavior included in the sequence of behavior performed by the former. One goal may be included in several goals and one goal may include several goals.
Extension	One goal is extending another goal by introducing alternatives or exceptions at a given extension point, respecting a given condition.
Generalization	One goal is generalizing another goal mean that the latter –the child – contains all the behavior and extension points defined in the former goal – the parent – and inherits all its used operators.
Equivalence	Two goals are equivalent when they serve the same purpose. Equivalence means the same definition for two or more goals. Equivalence could be functional and contextual, which means that the two goals perform the same behavior in the same context, or only functional which means that the two goals have the same behavior in two different contexts.
Precedence	One goal is preceding another goal means that the latter is sequenced to the behavior of the former.

4.3.7 Suggestion Engine

Another great usefulness of the goals patterns (section 4.2 and 4.3.3) is the suggestion system. In fact, based on their profiles described by the age, the types of goals (work, leisure or both) they are interested in, the areas of interest and the physical environment, end users will be guided in the process of goal/service composition through the database of goals patterns that contain the possible links (semantic functional, contextual and goal oriented information) between the various goals.

As goals patterns are created, improved and enriched by end users themselves, the suggestions will also be enriched and made more efficient. Our suggestion model is similar to e-mail interfaces - ex. Gmail; when writing a new message, and when the first recipient address is entered by the user, other addresses are proposed and suggested at the basis of the previous messages sent by this user.

4.3.8 Protocol and Query Language

The major role of this layer is to propose a query language that allows end users to express their needs and to describe their future goals. This protocol and query language should be very intuitive and user-friendly, but also powerful and rich in order to make rich, specific and precise description of the goals needed by end users. The goal query language allows querying, adding, modifying and removing stored goals. As with SQL for structured databases or SPARQL for the semantic web, the protocol and goal query language allows expressing interrogative and constructive requests using query constructs as SELECT, FROM, WHERE, ORDER, FILTER, etc. In addition to these features, the protocol and goal query language is able, if the request fails, to define the sub-goals that, combined, respond to the end user need. The protocol and goal query language should be able to determine an ordered set of sub-goals that will be transmitted to the composition language in order to compose the sub-goals and create a new goal.

It is necessary to emphasize that the protocol and query language is to be used by end users that have a clear idea of their overall goal and who can formulate it; other users who need help to set their goals can use the suggestions provided by the platform (see section 4.3.7).

4.3.9 Trust

To overcome the security and privacy problems, it is very important to focus on the enterprises governance; enterprises managers must have a clear policy towards the use of new technologies and create a strategy allowing the secure and successful adoption of a Web of goals platform. The strategies adopted should take place within the “Enterprise Infrastructure” and the layers of the proposed Web 4.0 architecture. Barhamgi, Benslimane, Ghedira, Tbahriti and Mrissa (2011) proposed a declarative approach to automatically combine data taking into consideration the data privacy constraints deduced from privacy policies, which determine the services that could be created by each role.

4.3.10 User Interface and Applications

A user interface, through which the end user interacts with the system, is a very important component. Our user interfaces owe to be smart, intuitive and user-friendly in order to guide end users during the process of goals creation through the use of the semantic. In fact, in order to compose goals, end users use their knowledge consisting on the objective of the goal, the final result, the frequency, the degree of importance, the duration, etc. This end-user knowledge represents the semantic which, alone, should be involved in the interaction between the end user and the Web of goals platform. Indeed, the service-to-service interaction, which is based on the syntax, is not valid at the interface level. The interface provides graphical display of services (called gadget) that represent sub-goals, which is an abstraction of services; therefore, the interaction and communication way at the interface level should also be an abstraction of the communication way between services and should rely on the semantic.

In addition to the use of semantic information, using forms is the easiest way for end users to interact with interfaces. In fact, Sommerville (2010) defines five primary styles of user interaction design, among which the forms that provide simple data entry and that are easy to learn.

5 CONCLUSION AND FUTURE WORK

In this paper we presented a vision for a new smart Web that will be called Web 4.0 or Web of goal, and which will have the mission of enhancing and making smarter our life in different aspects. We then proposed an architecture that will allow the new generation of end users (Web 4.0 end users) to be more autonomous and creative and to respond quickly to their own needs. Our new Web 4.0 architecture provides new principles and ideas that will accompany end users in the process of achieving their different goals; our architecture considers two types of end users: those who have a clear understanding of their goal, who can express it using a query language, and those who rely on a suggestion engine to discover the way leading them to their goal. Our architecture proposes also a goal description model, a goal composition model, a goal patterns, a logic engine, a goal context adaptation, trust, and user interfaces and applications.

Our future work consists of exploring the enabling technologies of our Web 4.0 architecture, in order to build a Web 4.0 operating framework.

REFERENCES

Aghaei, S., Nematbakhsh, M. A. and Farsani, H. K.

(2012). Evolution of the World Wide Web: From Web 1.0 to Web 4.0. *International Journal of Web & Semantic Technology (IJWesT)* Vol 3, N 1, January 2012.

Alberto, A. A., & Troutman, A. C. (2003). Applied behavior analysis for teachers. (*6th edition). Prentice Hall; 6 edition.* May 9, 2002. ISBN-13: 978-0130993878.

Anind K. D. (2001). Understanding and Using Context". Personal and Ubiquitous Computing. *Journal of Personal and Ubiquitous Computing, 2001,* Vol5, N1:4-7.

Barhamgi, M., Benslimane, D., Ghedira, C., Tbahriti, S. and Mrissa, M. (2011). A Framework for Building Privacy-Conscious DaaS Service Mashups. ICWS 2011. In *Proceedings of the 9th International Conference on Web Services*, DC, USA. 2011.

Benhaddi, M., Baina, K. and Abdelwahed, E. (2012). A user centric Mashuped SOA, *International Journal of Web Science (IJWS), Vol 1*, N 3, 2012: 204-223. DOI: 10.1504/IJWS.2012.045812.

Benhaddi, M., Baina, K. and Abdelwahed, E. (2013). Formalization of The User-Centric SOA Approach. In Proceedings of *the 15th International Conference on Enterprise Information Systems, Angers, France, Juillet 4-7, 2013.*

Bernardo, M. and Inverardi, P. (2003). Formal Methods for Software Architectures. Third International School on Formal Methods for the Design of Computer, Communication and Software*. Springer Editor. ISBN-13: 978-3540200833. 2003.*

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. *Scientific American. May 2001. p. 29-37.*

Bolchini, D. and Mylopoulos, J. (2003). From Task-Oriented to Goal-Oriented Web Requirements Analysis. In Proceedings of the *Fourth International Conference on Web Information Systems Engineering (WISE'03).*

Choudhury, N. (2014). World Wide Web and Its Journey from Web 1.0 to Web 4.0. *International Journal of Computer Science and Information Technologies (IJCSIT), Vol 5, N 6, 2014, 8096-8100*.

Clayton L. (1990). A research agenda for the nineties in human-computer interaction. *Journal of Human-Computer Interaction Vol 5, N 2. June 1*990. 125-143.

Cypher, A. (1993). Watch What I Do: Programming by Demonstration. The MIT Press, Cambridge. May, 4, 1993. ISBN-13: 978-0262032131.

Gawade, S.S. and Chhajed, G.J. (2014). Using Feedback Sessions for Inferring User Search Goals. *International Journal of Advanced Research in Computer and Communication Engineering Vol 3,* N 6, June 2014.

Hohpe, G. and Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional. October 20, 2003. ISBN-13: 978-0321200686.

Kritikos, K., Plexousakis, D., and Paterno, F. (2014). Task Model-Driven Realization of Interactive Application Functionality. *Journal of ACM Transactions on Interactive Intelligent Systems,* Vol 3, N4. January 2014.

Lankhorst, M. (2012). Agile Service Development: Combining Adaptive Methods and Flexible Solutions. Springer, April 2012. ISBN : 978-3-642-28188-4.

Liu, H., Liberman, H. and Selker, T. (2002). GOOSE: A Goal-Oriented Search Engine With Commonsense. In Proceeding of *the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Malaga, Spain,* May 29-31, 2002.

Liu, X., Huang, G., Zhao, Q., Mei, H. and Blake, M.B. (2014). iMashup: a mashup-based framework for service composition. *Journal of Science China Information Sciences. Science China Presss.* 2014, Vol 57, N 1. 1-20. ISSN : 1674-733X.

Ma, Y., Lu, X., Liu, X., Wang, X and Blake, M.B. (2013). Data-driven synthesis of multiple recommendation patterns to create situational Web mashups. Journal of Science China Information Sciences. 2013, Vol 56, N 8. 1-16. ISSN : 1674-733X.

Mehandijev, N., Lecue, F., Wajid, U. and Namoun, A. (2010). Assisted Service Composition for End Users. In *Proceedings of the 2010 Eighth IEEE European Conference on Web Services, Ayia Napa, 2010:131-138.*

Miles, R. and Hamilton, K. (2006). Learning UML 2.0. O'Reilly Media; 1 edition. May 5, 2006. ISBN: 978-0596009823.

Nedeve, V. and Dineva, S. (2012). New learning innovations with Web 4.0. In *Proceeding of the 7th International Conference on Virtual Learning ICVL, Nov 2-3,2012.* Brasov, Romania.

Nielsen, J. (1994). Goal Composition: Extending Task Analysis to Predict Things People May Want to Do. Accessible at: https://www.nngroup.com/articles/goal-composition/ (Last visit: Feb,19th, 2017).

O'reilly, T. (2007). What is Web 2.0 – design patterns and business models for the next generation of software. Communications & Strategies, *First Quarter 2007, N1: 17.*

Paterno, F. (2001). Task models in interactive software systems. In S.K. Chang, editor, Hand-Book of Software Engineering and Knowledge Engineering, 817-836. World Scientific Publishing Co., 2001. ISBN: 981- 02-4973-X.

Paterno,F. (2003). ConcurTaskTrees: An Engineered Notation for Task Models, Chapter 24, in Diaper, D., Stanton, N. (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction,* pp.483-503, Lawrence Erlbaum Associates, Mahwah.

Rose, D. E. and Levinson, D. (2004). Understanding User Goals in Web Search. In *Proceeding of the 13th international conference on World Wide Web*, May 17–22, 2004, New York, New York, USA.

Rui, K. and Butler, G. (2003). Refactoring Use Case Models: The Metamodel. In *Proceeding of the 25th Australasian Computer Society Conference (ACSC 2003).*

Shadbolt, N., Hall, W. and Berners-Lee, T. (2006). The Semantic Web Revisited. Intelligent Systems, *IEEE Intelligent Systems, Vol 21,* N. 96-101, May/June 2006, doi:10.1109/MIS.2006.62.

Sommerville, I. (2010). Software Engineering. *Addison-Wesley Publisher. March 13, 2010.* ISBN-13: 978-0137035151 | Edition: 9.