ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Evaluation of Addict Activities Trust in Cloud Computing

Patchava Vineela
M.Tech Student,
Nimra College of Engineering and Technology.

G.Minni
Guide,
Nimra College of Engineering and Technology.

Abstract:

Knowledge about computer users is very beneficial for assisting them, predicting their future actions or detecting masqueraders. In this paper, a new approach for creating and recognizing automatically the behavior profile of a computer user is presented.

In this case, a computer user behavior is represented as the sequence of the commands she/he types during her/his work. This sequence is transformed into a distribution of relevant subsequences of commands in order to find out a profile that defines its behavior.

Also, because a user profile is not necessarily fixed but rather it evolves/changes, we propose an evolving method to keep up to date the created profiles using an Evolving Systems approach. In this paper, we combine the evolving classifier with a trie-based user profiling to obtain a powerful self-learning online scheme.

We also develop further the recursive formula of the potential of a data point to become a cluster center using cosine distance, which is provided in the Appendix. The novel approach proposed in this paper can be applicable to any problem of dynamic/evolving user behavior modeling where it can be represented as a sequence of actions or events. It has been evaluated on several real data streams.

INTRODUCTION:

There exist several definitions for user profile. It can be defined as the description of the user interests, characteristics, behaviors, and preferences. User profiling is the practice of gathering, organizing, and interpreting the user profile information. In recent years, significant work has been carried out for profiling users, but most of the user profiles do not change according to the environment and new goals of the user. An example of how to create these static profiles is proposed in a previous work.

We propose an adaptive approach for creating behavior profiles and recognizing computer users. We call this approach Evolving Agent behavior Classification based on Distributions of relevant events (EVABCD) and it is based on representing the observed behavior of an agent (computer user) as an adaptive\ distribution of her/his relevant atomic behaviors (events).Used to monitor, analyze, and detect abnormalities based on a time-varying behavior of same users and to detect as queraders. It can also be applied to other type of users such as users of e-services, digital communications, etc.

STUDY OF THE SYSTEM:

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

1.Administrative user interface.

2.The operational or generic user interface.

The 'administrative user interface' concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The 'operational or generic user interface' helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities.

INPUT & OUTPOUT REPRESENTETION:

Input design is a part of overall system design. The main objective during the input design is as given below:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording.
- Data transcription.
- Data conversion.
- Data verification.
- Data control.
- Data transmission.
- Data validation.
- Data correction.

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input.
- Flexibility of format.
- Speed.
- Accuracy.
- Verification methods.
- Rejection rates.
- Ease of correction.
- Storage and handling requirements .
- Security.
- Easy to use.
- Portability.

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive.

As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

OUTPUT DESIGN:
In general are:

- External Outputs whose destination is outside the organization.
- Internal Outputs whose destination is with in organization and they are the User's main interface with the computer. Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs
- Operational outputs whose use is purely with in the computer department.
- Interface outputs, which involve the user in communicating directly with the system.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

OUTPUT DEFINITION:

The outputs should be defined in terms of the following points:

- Type of the output.
- Content of the output.
- Format of the output.
- Location of the output.
- Frequency of the output.
- Volume of the output.
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

- Will decimal points need to be inserted.
- Should leading zeros be suppressed.

OUTPUT MEDIA:

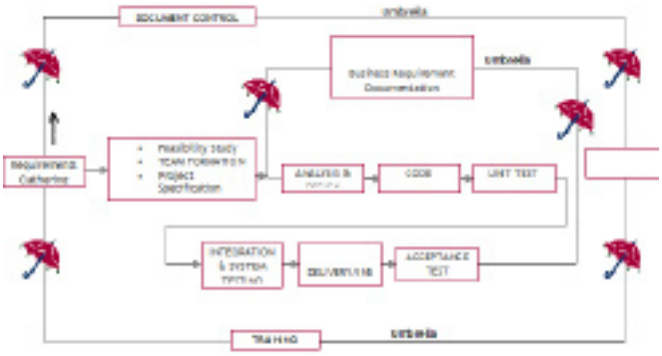
In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy.
- The response time required.
- The location of the users
- The software and hardware available.

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

2.3 PROCESS MODEL USED WITH JUSTIFICATION:

mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



SDLC (Umbrella Model):

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

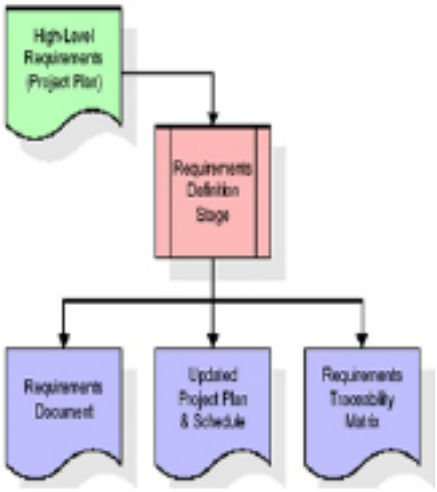
Stages in SDLC:

- Requirement Gathering.
- Analysis .
- Designing.
- Coding.
- Testing.
- Maintenance.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal.

In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

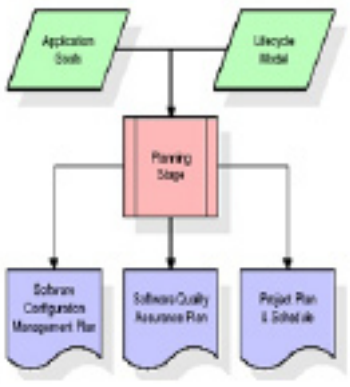
- Feasibility study is all about identification of problems in a project.

- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

- Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

Analysis Stage:

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



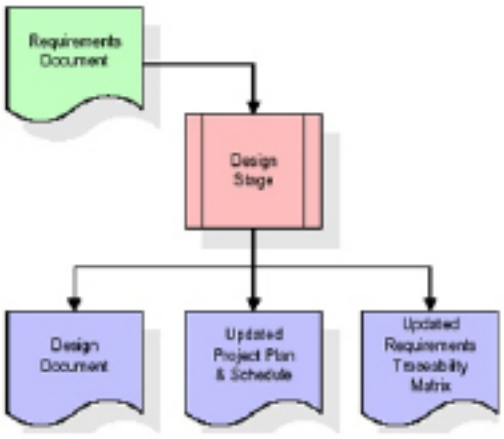
The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

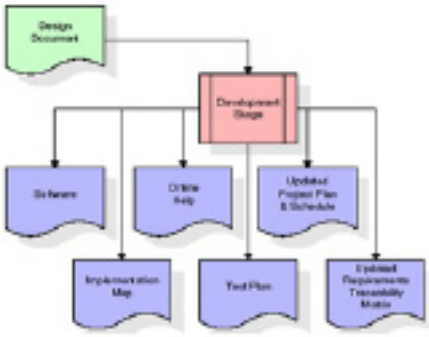


When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced.

Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

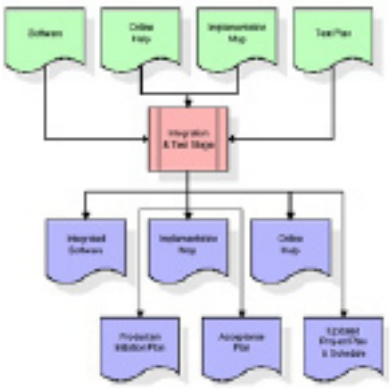


The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.
During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

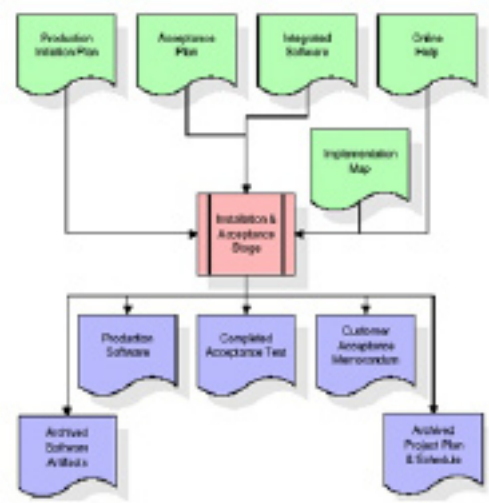
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

• Installation & Acceptance Test:

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

Maintenance:

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will under go training on that particular assigned category.

For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

2.4 SYSTEM ARCHITECTURE:

Architecture flow:

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tire architecture.

URL pattern represents how the requests are flowing through one layer to another layer and how the responses are getting by other layers to presentation layer through server in architecture diagram.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

REFERENCES:

[1] D. Godoy and A. Amandi, "User Profiling in Personal InformationAgents: A Survey," Knowledge Eng. Rev., vol. 20, no. 4, pp. 329 361,2005.

[2] J.A. Iglesias, A. Ledezma, and A. Sanchis, "Creating User Profilesfrom a Command Line Interface: A Statistical Approach," Proc.Int'l Conf. User Modeling, Adaptation, and Personalization (UMAP),pp. 90 101, 2009.

[3] M. Schonlau, W. Dumouchel, W.H. Ju, A.F. Karr, and Theus,"Computer Intrusion: Detecting Masquerades," Statistical Science,vol. 16, pp. 58 74, 2001.

12[4] R.A. Maxion and T.N. Townsend, "Masquerade Detection Using Truncated Command Lines," Proc. Int'l Conf. Dependable Systems and Networks (DSN), pp. 219 228, 2002.

[5] A. Alaniz Macedo, K.N. Truong, J.A. Camacho Guerrero, and M. Graca Pimentel, "Automatically Sharing Web Experiences through a Hyperdocument Recommender System," Proc. ACM Conf. Hypertext and Hypermedia (HYPERTEXT '03), pp. 48 56, 2003.

[6] D.L. Pepyne, J. Hu, and W. Gong, "User Profiling for Computer Security," Proc. Am. Control Conf., pp. 982 987, 2004.

[7] D. Godoy and A. Amandi, "User Profiling for Web PageFiltering," IEEE Internet Computing, vol. 9, no. 4, pp. 56 64, July/Aug. 2005.

[8] J. Anderson, Learning and Memory: An Integrated Approach. John Wiley and Sons, 1995.

[9] Y. Horman and G.A. Kaminka, "Removing Biases in Unsuper vised Learning of Sequential Patterns," Intelligent Data Analysis, vol. 11, no. 5, pp. 457 480, 2007.

[10] T. Lane and C.E. Brodley, "Temporal Sequence Learning and Data Reduction for Anomaly Detection," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 150 158, 1998.

[11] S.E. Coull, J.W. Branch, B.K. Szymanski, and E. Breimer, "Intrusion Detection: A Bioinformatics Approach," Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 24 33, 2003.

[12] P. Angelov and X. Zhou, "Evolving Fuzzy Rule Based Classifiers from Data Streams," IEEE Trans. Fuzzy Systems: Special Issue on Evolving Fuzzy Systems, vol. 16, no. 6, pp. 1462 1475, Dec. 2008.

[13] M. Panda and M.R. Patra, "A Comparative Study of Data Mining Algorithms for Network Intrusion Detection," Proc. Int'l Conf. Emerging Trends in Eng. and Technology, pp. 504 507, 2008.

[14] A. Cufoglu, M. Lohi, and K. Madani, "A Comparative Study of Selected Classifiers with Classification Accuracy in User Profiling," Proc. WRI World Congress on Computer Science and Information Eng. (CSIE), pp. 708 712, 2009.

[15] R. Polikar, L. Upda, S.S. Upda, and V. Honavar, "Learn++: An Incremental Learning Algorithm for Supervised Neural Net works," IEEE Trans. Systems, Man and Cybernetics, Part C (Applications and Rev.), vol. 31, no. 4, pp. 497 508, http:// dx.doi.org/10.1109/5326.983933, Nov. 2001.

[16] D. Kalles and T. Morris, "Efficient Incremental Induction of Decision Trees," Machine Learning, vol. 24, no. 3, pp. 231 242, 1996.

[17] F.J. Ferrer Troyano, J.S. Aguilar Ruiz, and J.C.R. Santos, "Data Streams Classification by Incremental Rule Learning with Para meterized Generalization," Proc. ACM Symp. Applied Computing (SAC), pp. 657 661, 2006.

[18] J.C. Schlimmer and D.H. Fisher, "A Case Study of Incremental Concept Induction," Proc. Fifth Nat'l Conf. Artificial Intelligence (AAAI), pp. 496 501, 1986.

[19] P.E. Utgoff, "Id5: An Incremental Id3," Proc. Int'l Conf. Machine Learning, pp. 107 120, 1988.

[20] P.E. Utgoff, "Incremental Induction of Decision Trees," Machine Learning, vol. 4, no. 2, pp. 161 186, 1989.