CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 70 (2015) 107 – 113
Procedia Computer Science

4th International Conference on Eco-friendly Computing and Communication Systems, ICECCS 2015

Dynamic Interactive Voice Response System Using Ontology and Java Expert System Shell

Thirumaran M.[a,*], Banupriya P.[a]

[a]*Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India*

Abstract

Interactive Voice Response (IVR) System is a technology that permits automated technologies to interact by way of customers via voice or Dual Tone Multi Frequency (DTMF) signalling keypad. An IVR system (IVRS) receives a combination of speech telephone input and keypad choice and affords suitable responses in the custom of voice. This paper is about the Dynamic IVRS for service oriented applications. In this system the request and response are handled as a services. The services can co-operate with each further to convey sophisticated added-value services. Ontologies are designed to maintain the insufficient information, hidden facts, knowledge sharing and also to handle the complete service functionalities. An ontology is a depiction (like a proper specification of a program) of the concepts and relationships that can exist for a customer. This system uses Java Expert System Shell for dynamic menu generation. JESS is a rule-engine and scripting environment which has the capability to context analysing by its intelligence. The objective of this paper is to reduce the execution delay which is measured in terms of the call length. This research optimizes the search space using Finite State Machine (FSM). It leads to better utilization and virtualization of IVR system.

© 2015 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015

Keywords: Interactive Voice Response System ; Dynamic ; Finite State Machine ; Context-based ; Semantic Web Services ; JESS;

1. Introduction to IVR

Call centers are now a days used by most of the organization to maintain association with the customers. Classic

* Corresponding author. Tel.: +91 9894592267.
E-mail address: thirumaran.pec@gmail.com

1877-0509 © 2015 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015
doi:10.1016/j.procs.2015.10.049

call center is structured to offer service to the user which may contrast for gathering information from simple to complex query. The complete communication maintained by the fully automated system is called Interactive Voice Response (IVR) system. The existing IVR structure are exact stationary in nature. The IVR structure shortage the awareness of the consumers who use the IVR scheme and don't consume reasoning ability to produce menus (or) choices based on the perspective and wants of the customer. Furthermore, the systems consume very a lesser amount of degree of failure recapture. Dynamic Interactive Voice Response System (IIVR) is an allowance of the outdated IVR system. The IVR system is Dynamic for the reason that it records the repetitive events of the customer and realizes the customer's requirement. The Java Expert System Shell (JESS) which is used as expert system which supports to craft rules which will be stated by the IVR system to make menus with dynamism.

The IVR systems are categorized into two types. They are Content-centred IVR and Service- centred IVR. In the Content-centred IVR scheme, the data directly fetches from the database and delivers it to the customers. In Service-centred IVR scheme, The IVR uses web services to extract info from the database. In this paper we proposed FSM based voice detected intelligent IVR algorithm. Finite State Machine (FSM) has set of states and transition function which transmutes to the subsequent state based on the input symbols which optimizes the search space. If any unsuccessful process during the request is handled by the IVR system, it restarts the states where it is stopped. This investigate optimizes the search space using Finite State Machine (FSM). Ontologies are deliberate to retain the inadequate information, hidden facts, knowledge sharing and also to treated the complete service functionalities.

2. Related Works

Due to the issue of system dependent API, shortage dynamic menu and degree of failure recover. Using of Expert system and semantic WSC Intelligent Interactive Voice Response System[1] proposed for context based dialogue menu. By presenting more necessary service and by decreasing the communication time the proposed system improves the quality of the service. Enhanced request-reply dialogue process achieved by use of JESS[1]

To modify the existing Rule Engine and without modifying the facts the Rule Engine is used which has the drawbacks of performance consumptions. To crack this concern author suggests three methods[2]. The methods are detail rule file content, sub function package division and compile out of process. When the user requirement changes the rule based system is changeable and maintainable depends on the performance general systems and rule-based systems is compared.

Due to the lack of unbalanced information the current IVR system does not provide customer satisfaction effectively. Author found a set of composed services by using dynamic IVR system. This mechanism[3] makes the user service extra speedy, trust worthy and precise. The (Fuzzy Analytical Hierarchical Process) FAHP which resolve the adjunct which remains between the composed set of service. The request-reply dialogue process between the service requestor and the service responder is used by the FAHP.

Most of the services are available without semantic description. So due to that service discovery fails to pertain a user related services. In proposed system developed semantic based web service discovery approach[4] which follows semantic based service enhancement and categorization. Service is accurately classified from the service functionality by the clustering methodology. The enhanced form of the service request is efficiently matched with the recovered service composition by applying Latent Semantic Indexing (LSI)[4].

Web service composition is the methodology which generate fresh service from the services available in the service repository. Mostly the web services are replicated with the same operation. Innovative method proposed for redundant-free web service composition[5]. They proposed the composition list method which follows a linked list data structure to choose the best service. They used Available Output Parameters (AOP) which uses a hash table to catch and remove the redundant web services from the development web service composition.

In IVR system the menus are generated dynamically based on Statistical Regression Algorithm[6] and it is implemented by IVR application gateway. The IVR server, the application gateway and the TTS (Text to Speech) component build together for a given data request the user getting a voice based response. In system, improved performance, reduced load and satisfied of customer needs.

In Traditional IVR, a computerized telephone system, requires only access to a telephone from any location. Responses were made by pressing numbers on the telephone keypad those responses were automatically stored in a Microsoft Access data file. Here the IVR was viewed as the data-collection method[7]. So there is lack of quality in the IVR call.

IVR system provide new and efficient method of longitudinal data collection. But due to lack of quality of IVR

call it leads to loss of reputation. To overcome this issue Verma et al., propose a new component called data boundary checker[8]. In which numerical responses were quality-checked against preset ranges and data were automatically time- and date-stamped and entered into a spreadsheet.

The preceding IVR system does not gives the appropriate quality of service to the customer. This IVR system uses API that are inbuilt in the system and also the information are directly extracted from the repository and delivers to the user. Proposed context aware interactive voice response system and FSM[9] which produced dialogue based context. IVR menu dynamically generated based on the user needs. The IVR system becomes more reliable and robust by use of FSM.

Subset of service to be called dynamically to make up the web service is called dynamic web service selection. By using FSM[10] the system allows to invocation sequence of web service is proposed to measure the probability of the state lead to successful execution where the web service may defeated. Eigen vector calculation is the same to calculation of accumulated reliability and control method is recycled to give it powerfully.

3. Proposed work

The proposed Dynamic Interactive Voice Response is context based and service based IVR system. The system registers various actions when the user interact with the IVRS.

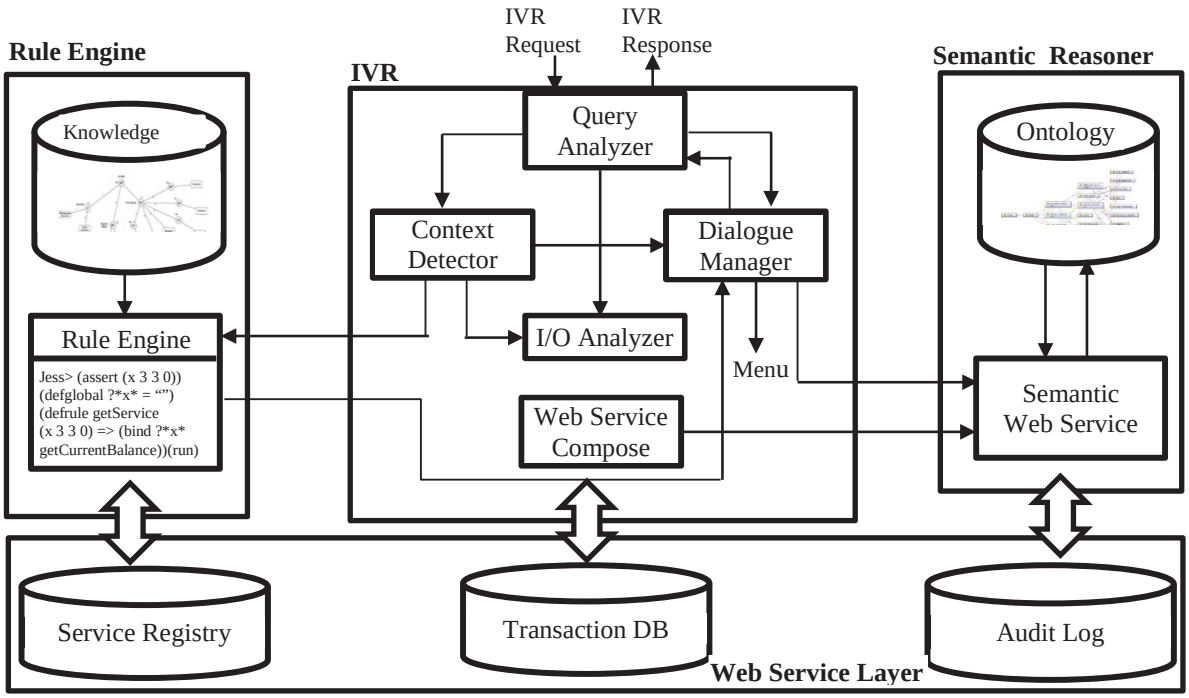

Fig. 1. Architecture of IIVR

In the earlier model, the degree of reoccurrence of the alternative and the options selected by them, the system try to recognize the user behaviour regarding IVR. Then the system dynamically produce menus depends on the context. The proposed architecture (Fig. 1) is developed by the three main components, they are Rule Engine, IVR System and the Semantic Reasoner.

3.1 IVR System

Initially when the user enters into IVR system set of default menu will be presented to them. Then the system get the request from the user and find whether the request is deterministic or non-deterministic. After user picking up the

options, Query analyzer structures the needs of the user based on the query inputted by the user and the context detector recognize the user context that is IVR sequence, request id and request time meaning how often does user use the same option. The output of the query analyzer's is inputted to the I/O analyzer for evaluating I/O requirement of the system for the particular operation.

3.2 Rule Engine

The responsibility of the rule engine is to analyze the order of dialogue flow from the result of the context detector. From the knowledge base system the JESS decides in which flow the dialogue should appear. Whenever the user interact with the system the knowledge base collects information about user behaviours and the knowledge keeps on updated. Based on the need of user requirement, the required rule is fetched from the JESS rule set and then it is extracted. If the rule does not available in JESS ruleset then the new rule will be created from the knowledge base. The dialogue manager creates dynamic menu based on the response from previous states and the user context when the system reached the user needs otherwise service composition will be performed from the available set of service.

3.3 Semantic Reasoner

The system not aware of how to progress the request or lack of knowledge about the request those are non-deterministic request which refer the semantic reasoner. It determines the I/O parameter with the support of the I/O analyzer. The web service composer fetched out the details for composing service from the ontology. If the service is subset of ontograph's vertex and edges then the service is mapped out to the ontograph or the service is bound with to the input and output parameter and then the required service is extracted and executed.

3.4 Web Service Layer

Service repository is the storage of all the services. Two main task done by the service repository, one is based on the user request the service is taken out from the service registry, another is the new composition service will be created from the service repository and the composed service also stored in service repository. All sort of transactions done by the user is stored in the transaction database. Audit log records the details about the option chosen by the user in the IVR system and session details of the user.

4. FSM Based Dialogue Manger Intelligent IVR Algorithm

4.1 FSM Diagram

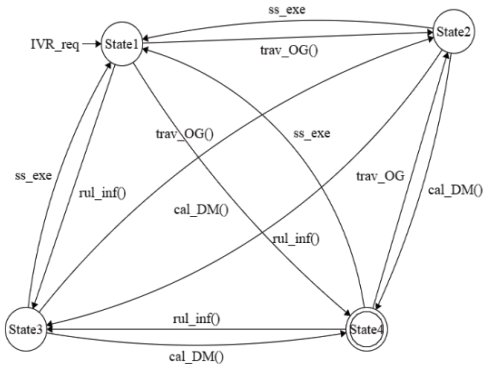

Fig. 2. FSM based Dialogue Manager

Table 1. State and Input description

State	Description	Input	Description
State1	Deterministic request	IVR_req() ss_exe()	Getting IVR request Simple Service execute
State2	Non deterministic request	trav_OG()	Traverse Ontograph
State3	Intelligent context analysing	Rul_inf()	Rule Inference
State4	Interface of IVR system	Cal_DM()	Call Dialogue Manager

4.2 Working Mechanism of IVR

Finite State Machine (FSM) has set of states and transition function which transforms to the next state based on the input symbols. In this paper we propose FSM based dialogue manager intelligent IVR algorithm which includes four states. Incase of any fail occur during the request is processed by the IVR system, it resumes the states where it is stopped

In state 1 after getting IVR request which is denoted by *req* from the user, the system analyses whether it belongs to single or compound request type which is denoted in the algorithm as *compound_req* and *single_req*. If the *req_type* is compound then directed to state 3. The direct service is represented as *direct[S],* If service can be directly fetched from the service registry and the request id is already exiting then the service will be invoked and executed that process is denoted as *lookup(serviceRegistry([S], s1))* and then the response (*res*) will be moved to state 4 for further process. If it is indirect service then it traverses ontograph (*call_traversal*).

4.3 Algorithm

State1:
req ← getIvrRequest();
req_type = compound_req || single_req;
if req_type == compound_req then
goto State3;
else If ((req == direct[S]) && (req_ID == known)) then
$s_1 \rightarrow$ filter(req);
lookup(serviceRegistry([S], s_1)) ;
where $\{s_1,s_2,\ldots s_n\} \in$ set of services S, s_1 is filtered service;
execute_s_1;
res ← out(s_1);
goto State4;
else
$s_1 \rightarrow$ call_traversal(Ontograph G, req)
goto State2;
end if
State2:
G→(Class, Subclass, Attr, Rel);
if $s_1 \subseteq$ G(V, E) then
mapping(s_1, G) = Class or Subclass;
goto State1;
else
$s_1 \rightarrow$ Boundwith($s_1[I_p]$, $s_1[O_p]$);
goto State1;
end if
State3:
RuleInference(IVR_sequence, req_id, req_time, context);
if r_1 match_with R where $\{r_1, r_2,..r_n\} \in$ JessRuleset R
then fetch r_1 from R;
extract_r_1;
res←compute(r_1), context[req]
where context[req] ← req_id || req_time || IVR_sequence;
goto State1;
end if
State4:
If reached user needs then
DialogueManager(res, context) → create DynamicMenu;
Else
$s_c \rightarrow$ serviceCompose;
execute_ s_c;
res ← out(s_c);
DialogueManager(res, context) → create DynamicMenu;
end if

If the service is subset of *G(V, E)* i.e ontograph's vertex and edges then the service is mapped out to the ontograph which is represented as *mapping(s1, G)* or the service is bound with to the input (I_p) and output parameter (O_p) and then it moves to state 1. In state 3 based on the user requirement, the required rule is fetched from the JESS rule set and extracted. The computational result of rule and the context request (*context[req]*) will be sent as a response to Dialogue Manager which creates dynamic menu based on the response from previous states and the user context otherwise service composition $_{Sc}$ will be performed from the available set of service.

5. Comparison to the state of art

Thus from the above Table 2 the traditional IVR is invariably static in nature and is not user friendly. The proposed IVR is compared with traditional with some of the parameters.

Table 2. Comparison of traditional and proposed IVR

Parameter	Traditional IVR	Proposed IVR
Type	Content based	Service based
User's Context	The dialogue does not change according to user's context	The dialogue changes according to user's context and needs
Failure Recovery	User is not notified about the failure	The session is maintained, next time when the user calls
Number of Dialogues	Fixed	Variable
Use of Domain-Specific Knowledge	Does not use	Uses domain-specific knowledge

6. Evaluation Metrics

The Evaluation metrics is used for estimating the proposed IIVR system with the current system are displayed below in Table 3. In the given table x denotes to a single web service; a indicates to Access Time; r_s mentions to Over-all amount of successful requests; rt refers to total amount of requests arrived; t_l refers to the total number of levels in the ontolograph; r_l refers to required level in the ontolograph.

Table 3. Evaluation metrics.

Metric	Formula	Explanation
Accuracy	$q_a = \left(\frac{r_s}{r_t}\right) x\ 100\%$	The time occupied by the IVR system to access data from the ontology
Knowledge Latency	$K_l = \left(\frac{r_l}{t_l}\right) x\ 100\%$	The time occupied to refer to the ontology by the IVR system that is lika a semantic reference

7. Experimental Result

Our proposed Dynamic Interactive Voice Response is set up to the banking application. Nearly we processed 200 request in the banking perspective and in which 98% of request successfully executed with the dynamism.

Table 4. Evaluation of the IVR system based on the performance metrics.

Request No.	Type	SET (*ms*)	SRT (*ms*)	RT (*ms*)	SR (*ms*)	Web servcie Composed /Instance used
1	I	3	2	9	7	S1 (Composed)
2	I	3	1.5	3	5	S1 (Instance)
3	D	2	1.5	7	9	-
4	I	2	1.5	2	5	S1 (Instance)
5	D	2	1.5	4	6	-
6	I	4	2	11	9	S1 (Composed)
7	I	3.5	2	6.5	7.5	S1 (Instance)
8	I	5	3	4	5	S1 (Instance)
9	D	2.5	2	7	8.5	-
10	I	2	2	6	8	S1 (Instance)

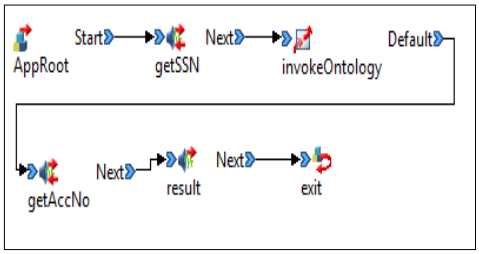

Fig. 3. Dialogue Designer.

The results are evaluated using the WAPT, Application Manager and other profiling tools. The results of the evaluation are tabulated below in table 4. The accuracy of the IVR system (qa) was found to be equal to 99.7%. The degree of failure recovery of the system (Df) was found to be = 98%. An IVR system was structured with the Avaya Dialog Designer plugin in Eclipse Galileo as shown in fig. 3 Which exhibits the process

of the dialogue manager. Text To Speech, announcement, connection and data are the tools used to design the dialogue manager.

Protege is a tool to design the Ontograph which consist of class, subclass, relationsphip between those classes and attributes and load at the IVR system is decreased by ontograph. The figure 4 shows the ontograph for the banking application which includes last deposit, Self employed are the sub classes of loan information, account information as class and the relationship between the classes are clearly mentioned. Figure 5 express the rules which is used in the banking application, the rule conditions are made though the use of JESS in the java platform. So in the rule engine, the rules should be already defined and can updated when a new (unknown) call flow occurs.

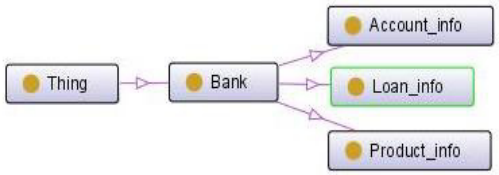

Fig. 4 Ontograph for banking.

```
Jess> (assert (x 3 3 0)) (defglobal ?*x* = “”)
(defrule getService (x 3 3 0) =>
 (bind ?*x* getCurrentBalance)) (run)

Jess> (assert (x 3 3 0)) (defglobal ?*x* = “”)
(defrule getService (x 3 3 0) =>
(bind ?*x* getLastDeposit)) (run)
```

Fig. 5 JESS.

8. CONCLUSION

Thus Dynamic Interactive Voice Response Systems performance a energetic part in improving customer interaction in an business organization. To overcome the drawbacks existences in content-centered IVR systems we examined and detect which were static we stepped towards IIVR system and henceforth a new method of service centered IVR beside with decision-making ability with the support of rule engine and semantic web services has been introduced. We then executed a domain-specific ontology, which incorporated with the existing services to produced dynamic menu. By make use of previous history, the frequent service used by the consumer is found and served to the rule engine for accurately display the required dynamic menu. Thus the proposed IVR system could improved the user fulfilment to a better degree than the existing content-based IVR system, reduce call length better utilization and virtualization of IVR system. Henceforth, this study optimizes the search space using Finite State Machine (FSM) and achieved the accuracy time was about 97.7% and degree of failure recovery was 98%.

References

1. Thirumaran M., Subham Soni , Brendha G. Gayathry. *An Intelligent Interactive Voice Response System for Banking Domain.* in Proc. Int. Conf. on Advanced Research in comput. Sci & Eng., 2015. doi: 10.1145/2743065.2743073.
2. Zhang G., Shan W. , Wang F. *Research on the Promotion of Rule Engine Performance*. In: Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on. pp. 1–3.
3. Thirumaran, M., P. Dhavachelvan, D. Aishwarya , K. Balachandar, 2012. *Dynamic and Optimal Interactive Voice Response System for Automated Service Discovery.* International Conference on Signal and Information Processing, Elsevier: pp: 146-152.
4. Paliwal A.V.,Shafiq B. , Vaidya J. Hui Xiong , Adam N., 2012. *Semantics-based automated service discovery*. Services Computing. IEEE Transactions., 5(2): 260-275.
5. Safi, A., D. Jawawi, K. Wakil., 2013. *Web services composition with redundancy consideration.* Open Systems (ICOS), IEEE: pp: 112-117.
6. Chang-Xing Qi , Jie Liu , Hang Li 2010. *Application research of a Statistical Regression algorithm in the IVR system*. Educational and Network Technology (ICENT), IEEE: pp: 358-360.
7. Lindsay JA , Minard CG , Hudson S , Green CE , Schmitz JM. *Using prize-based incentives to enhance daily interactive voice response (IVR) compliance: A feasibility study*. Journal of Substance Abuse Treatment, Volume 46, Issue 1, January 2014, Pages 74-77, ISSN 0740-5472
8. Verma SK , Courtney TK , Lombardi DA , Chang WR , Huang YH , Brennan MJ , Perry MJ. *Internet and telephonic IVR mixed-mode survey for longitudinal studies: choice, retention, and data equivalency*. Annals of Epidemiology, Volume 24, Issue 1, January 2014, Pages 72-74, ISSN 1047-2797
9. Thirumaran M., Subham Soni , Brendha G. Gayathry. *Design of Context-Aware Interactive Voice Response System*. on Advances in Natural and Applied Sciences, 2015, pp. 669-675.
10. San-Yih Hwang , Ee-Peng Lim , Chien-Hsiang Lee , Cheng-Hung Chen. Chen, 2008. *Dynamic web service selection for reliable web service composition*. Services Computing, IEEE Transactions., 1(2): 104-116.