Proceedings

2015 International Conference on Data and Software Engineering (ICoDSE 2015)

November 25th - 26th, 2015
Royal Ambarukmo Hotel,
Yogyakarta, Indonesia

IEEE Catalouge Number : CFP15AWL-ART

ISBN : 978-1-4673-8430-8

Department of
Computer Sciences and Electronics
Universitas Gadjah Mada

IEEE Advancing Technology for Humanity

ISBN: 978-1-4673-8430-8

Proceedings of

2015 International Conference on Data and Software Engineering (ICODSE 2015)

Universitas Gadjah Mada, Indonesia

November 25th – 26th, 2015

2015 International Conference on Data and Software Engineering (ICODSE 2015)

IEEE Catalog Number CFP15AWL-ART
ISBN 978-1-4637-8430-8

Additional copies of this publication are available from

Curran Associates, Inc
57 Morehouse Lane
Red Hook, NY 12571 USA
+1 845 758 0400
+1 845 758 2633 (FAX)
email: curran@proceedings.com

General Chair's Message

Welcome to 2015 ICODSE.
It is a great pleasure for us to host the 2015 International Conference on Data and Software Engineering (ICODSE 2015) at Yogyakarta, Indonesia. The ICODSE2015 conference aims at uniting researchers and professionals in the domains of data and software engineering, presenting and discussing high-quality research results and outcomes in their fields. This year, the conference is organized by Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. 2015 ICODSE is coorganized by Institut Teknologi Bandung and is also technically co-sponsored by IEEE Indonesia Section.

In total, we received 91 submissions of authors from 7 countries around the world. All submissions were peer-reviewed (blind) by at least three reviewers drawn from external reviewers and the committees, and as the result, 45 papers were accepted to be presented in this conference. These papers are in the proceedings of 2015 ICODSE.

Finally, as the General Chair of the Conference, I would like to express my deep appreciation to all members of the Steering Committee, Technical Programme Committee, Organizing Committee and Reviewers who have devoted their time and energy for the success of the event. For all participants, i hope you find the conference stimulating, fulfilling and enjoyable. I thank you for your support of ICoDSE2015 and your attendance, and wish you a pleasant experience in this conference and this beautiful city of Yogyakarta.

Aina Musdholifah
General Chair of the ICoDSE2015

Committee

General Chair

Aina Musdholifah	Universitas Gadjah Mada - Indonesia

Steering Committee:

Chair:

Edi Winarko	Universitas Gadjah Mada - Indonesia

Members:

A Min Tjoa	Vienna University of Technology – Austria
Sri Hartati	Universitas Gadjah Mada – Indonesia
Benhard Sitohang	Institut Teknologi Bandung – Indonesia
Iping Supriana Suwardi	Institut Teknologi Bandung – Indonesia
Khabib Mustofa	Universitas Gadjah Mada - Indonesia

Technical Program Committee

Chair:

Mhd. Reza Pulungan	Universitas Gadjah Mada –Indonesia

Members:

A Min Tjoa Vienna	University of Technology - Austria
Sri Hartati	Universitas Gadjah Mada - Indonesia
Wenny Rahayu	La Trobe University - Australia
Benhard Sitohang	Institut Technology Bandung - Indonesia
Jazi Eko Istiyanto	Universitas Gadjah Mada - Indonesia
Adila Alfa Krisnadh	Wright State University - USA

Agus Harjoko	Universitas Gadjah Mada - Indonesia
Aina Musdholifah	Universitas Gadjah Mada - Indonesia
Anny Kartika Sari	Universitas Gadjah Mada - Indonesia
Ary Setijadi Prihatmanto	Indonesia IEEE CS Chair - Indonesia
Azhari Sn	Universitas Gadjah Mada - Indonesia
Bayu Hendradjaya	Institut Teknologi Bandung - Indonesia
Dang Tran Khanh	Ho Chi Minh City University of Technology - Vietnam
David Taniar	Monash University - Australia
Dwi H. Widiyantoro	Institut Teknologi Bandung - Indonesia
Fathul Wahid	Universitas Islam Indonesia - Indonesia
I Wayan Mustika	Secretary, IEEE Indonesia Section - Indonesia
Khabib Mustofa	Universitas Gadjah Mada - Indonesia
Linawati	Universitas Udayana - Indonesia
Maguelonne Teisseire	University of Montpellier 2 - France
MM Inggriani	Institut Teknologi Bandung - Indonesia
Muhammad Ilyas Tariq	University of Sargodha - Pakistan
RV Hari Ginardi	Institut Teknologi Sepuluh November - Indonesia
Saiful Akbar	Institut Teknologi Bandung, Indonesia
Siti Zaiton Mohd. Hashim	Universiti Teknologi - Malaysia
Soleh Al Ayubi	Boston Children's Hospital - UnitedState
Tri Kuntoro Priambodo	Universitas Gadjah Mada - Indonesia
Vladimír Mařík	TU Prague - Czech
Wikan D. Sunindyo	Institut Teknologi Bandung - Indonesia
Yudho Giri Sucahyo	University of Indonesia - Indonesia

Organizing Committee

Afiahayati	Universitas Gadjah Mada - Indonesia
Andi Dharmawan	Universitas Gadjah Mada - Indonesia
Anny Kartika Sari	Universitas Gadjah Mada - Indonesia
Arif Nurwidyantoro	Universitas Gadjah Mada - Indonesia
Faizah	Universitas Gadjah Mada - Indonesia
M. Edi Wibowo	Universitas Gadjah Mada - Indonesia
Triyogatama Wahyu Widodo	Universitas Gadjah Mada – Indonesia

Reviewer

Arya Adriansyah	Eindhoven University of Technology - Netherland
A Min Tjoa	Vienna University of Technology - Austria
Abdiansah	Universitas Sriwijaya - Indonesia
AfiaHayati	Universitas Gadjah Mada - Indonesia
Agung Fatwanto	Uin Sunan Kalijaga - Indonesia
Agus Harjoko	Universitas Gadjah Mada - Indonesia
Aina Musdholifah	Universitas Gadjah Mada - Indonesia
Albertus Santoso	Universitas Atma Jaya Yogyakarta - Indonesia
Amin Anjomshoaa	Massachusetts Institute O Technology (Mit) - United States
Anastasia Widiarti	Universitas Sanata Dharma - Indonesia
Andi Dharmawan	Universitas Gadjah Mada - Indonesia
Anindita Septiarini	Univeristas Mulawarman - Indonesia
Anny Sari	Universitas Gadjah Mada - Indonesia
Anto Nugroho	Bppt - Indonesia
Arif Nurwidyantoro	Universitas Gadjah Mada - indonesia
Ary Noviyanto	University Of Auckland – New Zeland
Ary Prihatmanto	Institut Teknologi Bandung - Indonesia
Azhari Sn	Universitas Gadjah Mada - Indonesia
Bayu Hendradjaya	Institut Teknologi Bandung - Indonesia
Benhard Sitohang	Institut Teknologi Bandung - Indonesia
Bernard Suteja	Universitas Maranatha - Indonesia
Catur Rokhmana	Faculty Engineering
David Taniar	Monash University - Australia
Dhomas Hatta Fudholi	La Trobe University - Australia
Djati Mardiatno	Universitas Gadjah Mada - Indonesia

Dwi Widyantoro	Institut Teknologi Bandung - Indonesia
Edi Winarko	Universitas Gadjah Mada - Indonesia
Eka Karyawati	Universitas Undayana - Indonesia
Eko Sediyono	Universitas Kristen Satya Wacana - Indonesia
Emma Utami	Stmik Amikom Yogyakarta - Indonesia
Faizah	Universitas Gadjah Mada - Indonesia
Fathul Wahid	Universitas Islam Indonesia - Indonesia
Febriliyan Samopa	Institut Teknologi Sepuluh Nopember - Indonesia
Gasim	Stmik Global Informatika - Indonesia
Gerald Quirchmayr	University Of Vienna - Austria
Guruh Shidik	Universitas Dian Nuswantoro - Indonesia
Heru Ismanto	Universitas Musamus - Indonesia
Inggriani Liem	Institut Teknologi Bandung - Indonesia
Jazi Istiyanto	Universitas Gadjah Mada - Indonesia
Josef Küng	Johannes Kepler University Of Linz - Austria
Khabib Mustofa	Universitas Gadjah Mada - Indonesia
Kusrini Kusrini	Stmik Amikom Yogyakarta - Indonesia
Lilik Sumaryanti	Universitas Musamus - Indonesia
Linawati	Universitas Udayana - Indonesia
Lukman Heryawan	Universitas Gajah Mada - Indonesia
Maguelonneteisseire	Umr Tetis - Irstea
Mardhani Riasetiawan	Universitas Gajah Mada - Indonesia
Mayumi Kamada	Kyoto University - Japan
Michel Hassenforder	Uha – United States
Moh Edi Wibowo	Universitas Gajah Mada - Indonesia
Muhammad Tariq	Princeton University – United States
Putu Ciptayani	Politeknik Negeri Bali - Indonesia

Ratna Wardani	Universitas Negri Yogyakarta - Indonesia
Reza Pulungan	Universitas Gajah Mada - Indonesia
Roy Huizen	Stikom Bali - Indonesia
Rv Hari Ginardi	Institut Teknologi Sepuluh Nopember - Indonesia
Saiful Akbar	Institut Teknologi Bandung - Indonesia
Sheila Huda	Universitas Islam Indonesia - Indonesia
Sigit Priyanta	Universitas Gajah Mada - Indonesia
Siti Rochimah	Institut Teknologi Sepuluh Nopember - Indonesia
Soleh Al Ayubi	Boston Children's Hospital – United States
Sri Hartati	Universitas Gajah Mada - Indonesia
Sri Suwarno	Universitas Kristen Duta Wacana - Indonesia
Tb. Aimunandar	Universitas Serang Raya - Indonesia
Tri Priyambodo	Universitas Gajah Mada - Indonesia
Tran Khanh Dang	Ho Chi Minh City University of Technology - vietnam
Triyogatama Wahyu Widodo	Universitas Gajah Mada - Indonesia
Vladimir Marik	Czech Technical University - Prague
Wahyono	University Of Ulsan – South Korea
Wenny Rahayu	La Trobe University¬ - Australia
Wikan Sunindyo	Institut Teknologi Bandung - Indonesia
Wiranto Utomo	Universitas Kristen Duta Wacana - Indonesia
Yudho Sucahyo	Universitas Indonesia - Indonesia
Yudi Prayudi	Universitas Islam Indonesia - Indonesia
Yudistira Asnar	Institut Teknologi Bandung - Indonesia
Yunita Sari	Universitas Teknologi Petronas - Indonesia
Zainal Hasibuan	Universitas Indonesia - Indonesia

Contents

A Declarative Query Language Based on Speech Act Theory for Web Systems

Bambang Purnomosidi D. P.
Information Systems
STMIK AKAKOM
Yogyakarta, Indonesia
bdfl@bpdp.xyz

Lukito Edi Nugroho
Electrical Engineering and Information Technology
Gadjah Mada University
Yogyakarta, Indonesia
lukito@mti.ugm.ac.id

Paulus Insap Santosa
Electrical Engineering and Information Technology
Gadjah Mada University
Yigyakarta, Indonesia
insap@mti.ugm.ac.id

Widyawan
Electrical Engineering and Information Technology
Gadjah Mada University
Yigyakarta, Indonesia
widyawan@ugm.ac.id

***Abstract*—The World Wide Wide provides an enormous resources to be utilized by human. There are many possibilities of their existence, from unstructured data (can only be processed by human), semi structured data (structured data inside unstructured data), and fully structured data. In this context, many web scraper software exist to fetch and extract those data from the Web. However, until now, those tools are meant to just help human to fetch web pages / resources. Therefore, while useful for human, they do not significantly contribute to automation. To enhance level of automation, we must combine pragmatic capabilities into client / web scraper (as service consumer) and server / web application (as service provider). In this paper we propose a declarative query language for service consumer and service provider interaction. To enable pragmatic capabilities, we use illocutionary act from speech act theory as interaction patterns. The declarative query language reside on the service consumer's side while service provider provides resources to parse requests from service consumer and act accordingly. A prototype implementation of the declarative query language is presented.**

***Index Terms*—Query language, illocutionary act, speech act theory, web, web scraper, web interaction tool**

I. Introduction

Currently the World Wide Web (or just the "web") provides an enourmous resources and most of them are meant to be consumed by human. This situation is strictly normal since the web are developed mainly with human as its consumers. Unfortunately, human has limited capabilites, time, costs, and other constraints to utilize the web. Therefore, many efforts have been dedicated to make web more user friendly, not only for human but also for automatic processing by machine. According to [1], there are categories of web application if we based our category on web resources and how users interact with those resources. Those categories resemble human patterns in interaction and communication from linguistics knowledge domain: syntactic web, semantic web, and pragmatic web.

Syntactic web is used to categorize a web application which has no semantic / machine understandable contents, and no end point for automatic machine processing. This kind of web is not merely an old-time HTML-CSS-Javascript web application, but could be modern web application, as long as they can be consumed only by human then we may categorize it into syntactic web. **Semantic web**, coined by [2], is used to categorize a web application which provide machine understandable contents in the form of ontology, presented in standard way (using RDF and / or OWL2 for example). This kind of web application is good enough and one step ahead of web application improvement since it enables developers to create software agent for automatic and precise processing. **Pragmatic web** is getting more attention since so many machine understandable contents are available but they are context independent, therefore we need to shift to pragmatics realm which provides meaningful and contextual interaction. This kind of web systems, although has emerge more than decade ago [3], [4], is still quite challenging to be realized.

Different with semantic web which we can still call *semantic web application*, we may use term *pragmatic web systems* and not *pragmatic web application* alone since in a communication and interaction we must account for both parties: speaker and hearer (and both has speaker and hearer roles at once during conversation). Our intention in this paper is to improve web communication and interaction between client and service provider by mapping social layer into technology layer as depicted in Figure 1.

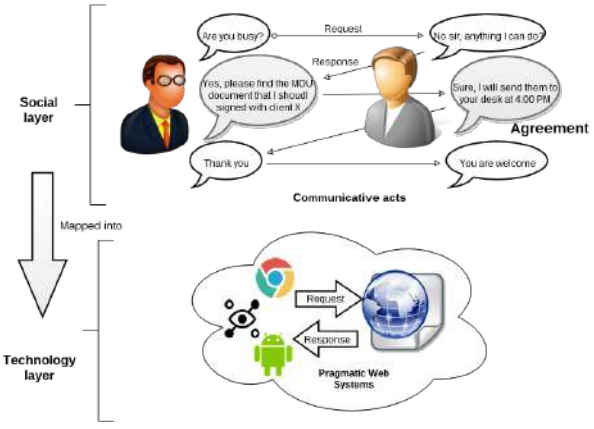

Fig. 1. Mapping of social layer to technology layer

It is necessary to also discuss pragmatic web from two perspectives: pragmatic web for limited web automation [3]

978-1-4673-8430-8/15/$31.00 ©2015 IEEE

and pragmatic web to augment human collaboration [4]. First perspective related with machine-to-machine interaction while second perspective is more related with providing web resources to augment human-to-human collaboration. In this case we should not advocate one perspective over the other one since both are correct but approach the word “pragmatic“ from different perspective. Pragmatic web for limited web automation is more influenced by (Distributed) Artificial Intelligence / Multi Agent Systems, therefore view web and internet as an environment for multi agent system with client and server / service provider as agents. Pragmatic web to augment human collaboration was emerged from LAP (Language Action Perspective). In LAP, contextual domain could not be rigidly defined since it needs sensitivity to context which more in the realm of human tacit knowledge and action. Our work in this paper is influenced by multi agent system research although at the same time we acknowledge LAP view, therefore we do not lay our view in this paper as fully agent-to-agent interaction to replace human but we use MAS research and knowledge domain to enable limited automation on the web. To this end, this paper will present a declarative query language for pragmatic interactions between client and server on the web systems.

This paper is structured as follows. Section II discuss previous related work in declarative query languages for web systems. Since our query language involves web resources, section III present a definition of web resources and an overview of web resources taxonomy which will become the object of our query language. This section is useful to help us define interaction possibilities. In section IV, we present speech act theory with emphasized in illocutionary act and then try to connect them with web interaction. In section V, we present an overview of declarative programming so that we may have a good foundation for the next section. In section VI, we integrate all of previous discussion into a specification of a declarative query language for web systems: IfA (Interaction for Action). We also present IfA reference implementation. We conclude our paper in section VII with an overview of what we try to solve and possible future research.

II. Related Work

Complexities and uncertainties in web systems have lead to structured and unstructured resources on the web. Developers and researchers have developed some query languages for web systems. Until recently, the most mature query language on the web is SPARQL[1]. SPARQL is used to query and manipulate RDF graph content on the web or in RDF datastore. It has similar construct with SQL. Some query languages are also based on RDF, such as Versa[2], iTQL[3], N3QL[4], and many others[5]. A more limited functionalities is offered by CQL[6] which only deal with web indexes, bibliographic catalogs, and museum collection information. A query language for Linked Data (LDQL), mainly utilize RDF, has also been developed [5].

Some query languages act as web scraping software and able to analyze contents of query result at once. Some examples of this category are Yahoo Query Language (yql[7]), WebSQL [6], WebOQL [7], Squeal [8], and HTQL[8]. XQuery can also be used as query language for web systems since it is meant to process XML contents which is common on the web.

Our approach is different. We do not merely consider web systems as static resources, instead we consider web systems as a system where client and server can communicate and interact dynamically by enforcing pragmatic interactions between client and server. Therefore, our proposal are meant for wider intention: it should enable query to static resources while also provide infrastructure for pragmatic interactions. In our paper, previous query language can always be accomodated using a component of illocutionary force: REQUEST to syntactic or semantic web resources.

III. Taxonomy of Web Resources

The term “resources“ is not common on the web in its early stage of development. Normally, web was created to share information using interlinked documents and files, therefore documents and files are the terms that common to denote “resources“. In this paper, we use the term “web resources“ as identified by RFC (Request for Comment) 3986[9]:

> ... the term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today’s weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

Discussion about “surface web“ (for indexed web resources by search engines) and / or “dark web“ (non-indexed web resources by search engines) is not within the scope of this paper. We consider those resources are known in this paper. Further, we concentrate mainly in open data on the web (“linked open data“) with one exception: for sensitive

[1] http://www.w3.org/TR/sparql11-overview/

[2] http://xml3k.org/Versa

[3] currently unavailable, was created as part of Kowari RDF store

[4] http://www.w3.org/DesignIssues/N3QL.html, a query language for RDF based on Notation3

[5] See A comprehensive list of RDF query language at https://en.wikipedia.org/wiki/RDF_query_language

[6] http://www.loc.gov/standards/sru/cql/

[7] https://developer.yahoo.com/yql

[8] http://htql.net

[9] https://tools.ietf.org/html/rfc3986

and private data / information, we use “https“ especially for “directives“ type of illocutionary acts.

A. Syntactic Web Resources

In this resources, we identify web resources as HTML-related contents suitable for human processing. This resources, eventhough available for human, can also be analyzed by machine with the help of DOM manipulation tools and HTML parser such as “jsdom“[10] and “htmlparser2“[11]. Beside HTML-related contents, there are also many “non machine understandable resources“ available on the Internet/WWW. They do not have special tagging to convey meaning. These resources are usually consist of proprietary content without semantic tagging such as *.doc, .xls, .psd,* etc.

B. Semantic Web Resources

Semantic web resources are available in many forms and those data are context independent, therefore we may scrap them “as is“ and have them available for later processing. An exception to this is “SPARQL endpoint“ since it provides an endpoint which can be used by developers to “ask“ questions. Semantic web resources are all web resources which is meaningful for machine so that machine can “understand“ the data. Search engine usually use semantic tagging to identify “meaning“ of data so that they can provide best answer when user search some contents based on some keywords. Search engine usually uses web crawler to crawl a website and then put data about them inside search engine store. By using semantically tagged content, crawler can mine the contents correctly and searching algorithms can be developed by search engine to provide users with a more precise answer. Note that while search engines can utilize them, they usually only tell users "here’s the data about XYZ", but they can not answer whether "Is data X availabe inside?".

To convey meaning into data, developers use vocabularies. The term “vocabularies“ is interchangeable with “ontologies“ and there is no clear division between both[12]. In this paper, both terms are used since some of them are already attached such as OWL (Web Ontology Language) and “vocab“ attribute in RDFa, but when one meet both terms in this paper, please note that they are interchangeable. To tag contents with specific meaning, developers can use commonly available vocabularies on the Internet (see http://schema.org) for example. When developers need to express more meaning and relationship between meaning then they can build their own vocabularies using “schema“ (RDFS - RDF Schema, XML schema, and Document Type Definition). Vocabularies alone are not enough, however. We still need serialization format to represent them. Original specification is XML but more serialization are available such as JSON, JSON-LD, N3, Turtle, N-Triples, etc.

[10]https://github.com/tmpvar/jsdom
[11]https://github.com/fb55/htmlparser2
[12]http://www.w3.org/standards/semanticweb/ontology

Currently, vocabularies or ontologies are described using RDF, RDFS, and OWL. They are all family of knowledge representation language. RDF (Resource Description Framework) is used to make a statement about resources by represent data in triple graph (subject-predicate-object). RDF is structured with RDFS (RDF Schema). OWL2 is used to create ontologies, mainly for ontologies which need complex expressive power such as semantic relationships. On top of RDF, RDFS, and OWL2, some common vocabularies exist, for example FOAF (Friend Of A Friend) and vocabularies described by schema.org. Those are all semantic web resources but we do not discuss all available vocabularies.

1) Stand Alone Semantic Web Resources: Developers may put ontologies on the Internet as a text file with specific serialization. For example, RDF/XML represent semantic data in XML format and those data can be put in a URI such as http://server/foaf.xml and therefore can be crawled and / or scrapped by agent. Other possibilities are:

- Plain XML data
- JSON-LD file
- JSON file

2) Triplestore and SPARQL Endpoint: Triplestore is software which can be used to manipulate RDF data. It usually provides SPARQL capabilities. SPARQL (*SPARQL Protocol and RDF Query Language*)[13] is a specification from the World Wide Web Consortium to retrieve and manipulate datasets stored using RDF. As of SPARQL 1.1, the query consists of:

- **SELECT** to extract raw data in a table format
- **CONSTRUCT** to extract information and transforms the result into valid RDF
- **ASK** to provide a “true“ or “false“ result from a query
- **DESCRIBE** to extract RDF graph
- **INSERT** to insert datasets into triplestore
- **DELETE** to remove datasets from triplestore

3) Embedded Semantic Resources Inside Non Semantic Resources / HTML: In this category, resources are embedded inside syntactic web resources / HTML. In this category, developers provide human processing related contents as well as machine understandable contents inside a resource. They both usually provide the same data / information, only with different processors. Some of them are:

- RDFa (Resource Description Framework in Attributes) provides a way to embed metadata using attribute inside (X)HTML tag. RDFa consists of RDFa core, RDFa Lite, XHTML+RDFa, and HTML5+RDFa. RDFa is W3C recommendation[14].
- Microdata uses attributes inside HTML5 tag to convey data semantic. It is not W3C recommendation but WHATWG HTML specification.
- SVG (Scalable Vector Graphics) is a specification from W3C to represent image using tags. Inside SVG, we may put metadata using RDF and vocabularies such as Dublin Core to describe the image semantically.

[13]http://www.w3.org/TR/sparql11-query/
[14]http://www.w3.org/TR/rdfa-primer/

- Microformats provides a way to use attribute to convey meaning. Microformats define some vocabularies which can be embed into HTML attribute. Some examples of microformats vocabularies are “h-card“, “h-calendar“, “rel-license“, “h-review“, etc.

C. Pragmatic Web Resources

Currently, we acknowledge that there is no canonical definition of “pragmatic web“ resources. In this paper, we define pragmatic web resources as resources which can be used for pragmatic (contextual) interaction. As such, we put responsibility of interaction to two parts: client and service provider (server). This kind of resources are our main concerns. For contextual interaction, both parties have to understand not only syntax and meaning (semantic), but also need to know intention of each party. Conceptually, client send data to service provider and service provider provides a URL to receive those data. When sending data, client need to express its intention. Service provider will receive data together with client intention and then process them and send response together with service provider's intention. Those interaction patterns are common in human everyday life. In this paper, we propose speech act theory for pragmatic interaction patterns.

IV. Speech Act Theory for Pragmatic Interactions

We express interaction between client and server as communicative act which in turn is inherited from Speech Act Theory [9], [10]. This theory is useful for communication between agents (agent as utterer and agent as hearer) and its effect to both parties in a structured way. In each interaction and communication, each side has intention and content of intention. Intention is known as *illocutionary force*, while content of intention is known as *propositional contents*. Our paper is developed based on formula 1.

$$\boxed{f(p)} \tag{1}$$

f represent *illocutionary force*
p represent *propositional contents*

Illocutionary force is based on illocutionary act in english verbs as depicted in Table I.

V. Declarative Programming for Interactions on the Web

Declarative programming is a programming paradigm that express the logic of computation without describing its control flow [11]. In this paradigm, developers develop program from a higher abstraction, i.e. they specifiy *what* is to be computed, not necessarily *how* it is to be computed [12]. Take SQL for example. Developers do not need to specify details of logic in programming language primitive to get all of the data inside a table (which maybe consists of some sequence: open file, search data, retrieve data, close file), insted they just write “SELECT * FROM tablename“.

Illocutionary act	Description	English verbs
Assertives	Can be verified as true or false	assert, claim, affirm, assure, inform, predict, report, suggest, insist, hypothesize, swear, admit, confess, blame, praise
Directives	Call upon listener to do something	direct, request, ask, urge, demand, command, forbid, suggest, insist, recommend, implore, beg
Commissives	Commit to a course of action	promise, vow, fledge, swear, consent, refuse, assure, guarantee, contract, bet
Expressives	Express a psychological position about a state of affairs	apologize, thank, condole, congratulate, complain, protest, compliment, praise, welcome
Declaratives	Change the reality in accord with the proposition of the declaration	fire, pronounce, declare, appoint, confirm, endorse, renounce, denounce, name, call, repudiate

TABLE I
Summary of English verbs related to illocutionary acts

Consider that web systems also consists of many resources, this kind of programming paradigm is suitable to build application which makes use of interaction between client and service provider. Therefore, in this paper we propose a declarative query language for web systems.

VI. IfA Language: Grammar and Reference Implementation

A. IfA Grammar

IfA (Interaction for Action) is a declarative query language which we develop to address our needs for a web interaction tool. To this end, we developed a grammar for IfA and then put some source code inside for programming logic. Using this approach, we do not put responsibilities of programming logic on developers, instead we help developers to specify their intention using this language and have our interpreter do the tasks. By using our IfA reference implementation, developers do not need to use a web browser and can use the interpreter (which is command line program / headless web client) combined with any general purpose programming language create a more powerful agent for web automation.

Our grammar was developed using PEG (Parsing Expression Grammar) [13]. There are tools available at Bryan Ford homepage[15] for PEG parsers, PEG parser generators, and / or combinator libraries which can be used to develop an interpreter based on our developed grammar. Listing 1 shows partial IfA grammar. Note that our original grammar is more comprehensive and include source code for programming logic, therefore we do not put full grammar here, only relevan parts (for example, in this paper we assume only GET method of HTTP for interaction, while our full specification is much more complete). Interested readers are invited to get the full grammar and the software at http://github.com/weberia/pithy.

[15]http://bford.info/packrat/

```
start
    =   (force EOL)+
force
    =   representatives / directives / commissives /
        expressives / declaratives

representatives
    =   assertive / informative
assertive
    =   'ASSERT' _ url:string _ propositional:string
        { assert(url, propositional) }
informative
    =   'INFORM' _ url:string _ propositional:string
        { inform(url, propositional) }

directives
    =   'REQUEST' _ url:string { getRequest(url) }

// (pragmatics) Making a commitment, such as a promise or
// threat, by illocutionary means.
//      ex: "I will buy if you give me 30% discounts"
commissives
    =   'CONSENT' _ url:string _ action:string _
        requirements:string
        { consent(action, requirements) }

expressives
    =   apologize / thank / complain
apologize
    =   'APOLOGIZE' _ url:string _ propositional:string
        { apologize(url, propositional) }
thank
    =   'THANK' _ url:string _ propositional:string
        { thank(url, propositional) }
complain
    =   'COMPLAIN' _ url:string _ propositional:string
        { complain(url, propositional) }

declaratives
    =   'DECLARE' _ url:string _ propositional:string

string "string"
    =   quotation_mark chars:char* quotation_mark { return
        chars.join(""); }
char
  =   unescaped
    / escape
    sequence:(
        '"'
      / "\\"
      / "/"
      / "b" { return "\b"; }
      / "f" { return "\f"; }
      / "n" { return "\n"; }
      / "r" { return "\r"; }
      / "t" { return "\t"; }
      / "u" digits:$(HEXDIG HEXDIG HEXDIG HEXDIG) {
          return String.fromCharCode(parseInt(digits,
              16));
        }
    )
    { return sequence; }
escape
    =   "\\"
quotation_mark
    =   '"'
unescaped
    =   [\x20-\x21\x23-\x5B\x5D-\u10FFFF]
HEXDIG
    =   [0-9a-f]i
EOL
    =   [\n\r]*
_ "whitespace"
    =   [ \t\n\r]*
```

Listing 1. IfA grammar, partially written

B. IfA Reference Implementation

We develop our IfA reference implementation using PEG.js[16]. The grammar should be easy to comprehend and ported to another tool such as Mouse[17]. To use this software, one need to have Node.js[18] installed first. Basically, what needs to be done is clone its repo and execute "pithy.js" using Node.js as follows:

```
$ git clone http://github.com/weberia/pithy
$ cd pithy
$ node pithy.js

  Usage: pithy <options>

  Options:

    -h, --help                output usage information
    -V, --version             output the version number
    -i, --inputfile <file>  Script file consists of IfA
        commands to be executed
$
```

Currently, a script consists of IfA queries should exists. This script is used as "pithy" argument. Execute "node pithy.js -i examples/test0.ifa" will read "examples/test0.ifa" file (relative to current directory) and have its contents executed.

C. Server Handler

On the server side, as previously explained, there are three possible parts:

- Syntactic resources: server provides HTML-based resources which could be any URL.
- Semantic resources: server may provides static route for static semantic resources (RDF files, OWL2 files, etc), RESTful web services with on-the-fly semantic resources, and SPARQL endpoint.
- Pragmatic resources: server provides a RESTful pragmatic route for interaction. We propose route in the form of "/illocutionary-force/params", for example "http://server/inform/params" to accept "INFORM" illocutionary force from client. However, logic for route handler is context dependent and its implementation is not in the scope of our paper.

D. Some Partial Query Examples

In these examples, we use "token" for transaction id and "id" for client id. In reality, token is an agreement between client and server. The "propositional" parts need to be agreed upon and expressed in a resource description format as part of service provider specification so that client knows how to handle interactions. Note that we use "ampersand - &" to denote parameters / arguments separation, the same with HTTP GET method.

```
REQUEST
    "https://provider/request"
    "id=CUST123&pass=pass123&contents=list-all-products"
```

Listing 2. REQUEST query

[16]http://pegjs.org
[17]http://www.romanredz.se/Mouse/index.htm
[18]http://nodejs.org

In listing 2, we send request to service provider. Our request is "list of all products". In return, we will have a token and list of all products in JSON formatted stream. This token will be used through out the examples. Next query, we can narrow our query to list a product specification with different token result.

```
ASSERT
    "http://provider/assert"
    "token=e9b76b473bbc51be2dec5d1a96630881&id=CUST123&
content=assert-buy"
```

Listing 3. ASSERT query

In listing 3, we send an ASSERT query to service provider that we buy the product with described token.

```
INFORM
    "http://provider/inform"
    "token=e9b76b473bbc51be2dec5d1a96630881&discounts=30&
id=CUST123&reasons=long-time-customer"
```

Listing 4. INFORM query

In listing 4, we inform service provider that we want "30 percent discounts" and as a consideration, we send information that we are long time customer of service provider (to be verified by service provider).

```
CONSENT
    "http://provider/consent"
    "token=e9b76b473bbc51be2dec5d1a96630881&id=CUST123&
action=buy&reqs=discounts0"
```

Listing 5. CONSENT query

In listing 5, we make a "promise" to service provider that we will buy the product. If service provider agree, then a new token will be sent and we should ASSERT that new token.

```
APOLOGIZE
    "http://provider/apologize"
    "token=e9b76b473bbc51be2dec5d1a96630881&id=CUST123&
content=not-deal"
```

Listing 6. APOLOGIZE query

In listing 6, we send a query that we apologize for something. This could be, for example, we apologize that service provider can not close a deal with us. This is strictly optional and is used to show that we are a "well-behaved" client.

```
THANK
    "http://provider/thank"
    "token=e9b76b473bbc51be2dec5d1a96630881&id=CUST123&
content=data-accepted"
```

Listing 7. THANK query

In listing 7, as a well-behaved client, we send "thank you" to service provider since we have received product specification.

```
COMPLAIN
    "http://provider/complain"
    "token=e9b76b473bbc51be2dec5d1a96630881&id=CUST123&
content=quality-unaccepted"
```

Listing 8. COMPLAIN query

In listing 8, we file a complain that product specification can not be accepted in terms of quality.

```
DECLARE
    "https://provider/declare"
    "type:buy&token=e9b76b473bbc51be2dec5d1a96630881&
id=CUST123"
```

Listing 9. DECLARE query

In listing 9, we declare that we have a deal with service provider to buy something. This kind of illocutionary act changes world state, in that case we become a customer and we have to send the money as our obligation, while service provider needs to send product(s) or conduct some services for us.

VII. Conclusion

In this paper, we present IfA, a declarative query language based on speech act theory for (pragmatic interactions) on the web. We acknowledged a taxonomy of web resources and design a language for client server interactions on the web based on those resources. We also made its reference implementation available so that interested reader can use this prototype and use it where possible to enhance their software agent. However, we realize that there will be many enhancements to be made for future research such as scaling its language syntax / commands and develop a language game for meaningful discourses on the web. As we understand, a meaningful discourses can not happen only with one interaction but a series of conversation with specific goals. Out paper provides a solid foundation for meaningful discourses between client and service provider on the web.

References

[1] K. Liu and A. Benfell, "Pragmatic web services: A semiotic viewpoint," in *Software and Data Technologies*, ser. Communications in Computer and Information Science, J. Cordeiro, A. Ranchordas, and B. Shishkov, Eds. Springer Berlin Heidelberg, 2011, vol. 50, pp. 18–32. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20116-5_2

[2] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[3] M. P. Singh, "The pragmatic web: Preliminary thoughts," in *Proc. of the NSF-OntoWeb Workshop on Database and Information Systems Research for Semantic Web and Enterprises*, 2002, pp. 82–90.

[4] M. Schoop, A. de Moor, and J. L. Dietz, "The pragmatic web: A manifesto," *Communications of the ACM*, vol. 49, no. 5, pp. 75–76, 2006.

[5] O. Hartig and J. Pérez, "Ldql: A query language for the web of linked data," in *Proc. 14th Int. Semantic Web Conf*, 2015.

[6] G. A. Mihaila, "Websql-an sql-like query language for the world wide web," Ph.D. dissertation, Citeseer, 1996.

[7] G. O. Arocena and A. O. Mendelzon, "Weboql: Restructuring documents, databases and webs," in *Data Engineering, 1998. Proceedings., 14th International Conference on*. IEEE, 1998, pp. 24–33.

[8] E. Spertus and L. A. Stein, "Squeal: a structured query language for the web," *Computer Networks*, vol. 33, no. 1, pp. 95–103, 2000.

[9] J. L. Austin, *How to do Things with Words. The William James lectures delivered at Harvard University in 1955.*, J. O. Urmson, Ed. Clarendon Press, 1962.

[10] J. R. Searle, *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969, vol. 626.

[11] J. W. Lloyd, "Practical advtanages of declarative programming." in *GULP-PRODE (1)*, 1994, pp. 18–30.

[12] O. Torgersson, "A note on declarative programming paradigms and the future of definitional programming," *Das Winteroete*, vol. 96, no. 1996, p. 13, 1996.

[13] B. Ford, "Parsing expression grammars: a recognition-based syntactic foundation," in *ACM SIGPLAN Notices*, vol. 39, no. 1. ACM, 2004, pp. 111–122.

Author Index

ICoDSE 2015
International Conference
on Data and Software Engineering

ICoDSE 2015

Data and Software Engineering in The Age of Social Media
Nov 25th - 26th, 2015 | Yogyakarta, Indonesia

CERTIFICATE

In appreciation of participation to **"THE 2015 INTERNATIONAL CONFERENCE ON DATA AND SOFTWARE ENGINEERING"**,

we present this certificate to

BAMBANG PURNOWOSIDI

as a **PRESENTER**

with the paper entitled: **A Declarative Query Language Based on Speech Act Theory for Web Systems**

Head of the Department of Computer Science
and Electronics, Universitas Gadjah Mada

EDI WINARKO, M.Sc., Ph.D.

General Chair Committee

ICoDSE 2015
International Conference
on Date and Software Engineering

AINA MUSDHOLIFAH, M.Kom., Ph.D.

Department of Computer Science and Electronics,
Faculty of Mathematics and Natural Science
Universitas Gadjah Mada

IEEE
Advancing Technology
for Humanity