From informal to formal knowledge representation using QAF triad

Eugeniu Costetchi and Eric Ras
Public Research Center Henri Tudor
29, avenue John F. Kennedy, L-1855 Luxembourg, +352 42 59 91 1
{Eugeniu.Costetchi and Eric.Ras}@tudor.lu

ABSTRACT
Knowledge acquisition is a tedious human intensive process. It has been, and still is a bottle neck in development of knowledge bases. Most of the solutions that have been proposed to this problem focus on static text processing and understanding. They avoid dealing with human interaction and "understanding" the meaning of human utterances on the fly because of the complexity. Many of the knowledge elicitation techniques are shaped or can be shaped as interviews – highly dynamic dialogue interactions. Having in mind the context of knowledge elicitation interview, we propose an approach to automatize such interaction process. We propose QAF triad as a knowledge acquisition instrument. It serves as interaction paradigm as well as a data structure to support human-machine written-dialogue interaction. It is also independent of the knowledge elicitation technique and formal language used.

Keywords
knowledge acquisition, human-machine interaction, QAF triad, dialogue system

1. INTRODUCTION
In this paper we propose a dialogue-based approach to elicit, conceptualize, and formalize the user knowledge. Knowledge acquisition from humans is considered to be more of an art than a science, most of the research is very pragmatic and there are not many attempts to formalize these techniques. An attempt has been made by Rugg when he proposes Method Fragments [1] as a controlled language to describe elicitation techniques. He showed that knowledge elicitation techniques can be recursively decomposed into fragments. However, Rugg gives no semantic description of such a framework. But what is the atomic construct that holds the semantics of an elicitation technique? We propose the concept of the Question-Answer-Fact triad (QAF triad) as the atomic "fragment" of an elicitation technique and we describe its semantics.

To motivate the QAF as an atomic fact, let us assume that that we can take it even to a lower level, i.e., to consider as atomic "fragments" the question and answer as separate and distinct entities, then it is a grand challenge to connect the two. Making such a connection requires deep language processing and understanding. Instead, if we do not dissociate the question and the answer then the burden of deep language analysis and understanding is relieved from the dialogue system. If the question and answer are treated holistically then, as a consequence, a third component emerges. It is the factual expression obtained from the imperative rephrasing of the question and its answer, i.e., the imperative form of the knowledge that has been obtained from the answer to the given question. This fact can be formulated in human understandable or machine processable forms. Because of its dual nature, this emerged fact serves as a translation pivot from informal to formal knowledge. Following the previous argumentation, we propose the QAF triad as the atomic "fragment" to be used in elicitation technique description.

In addition, one of the fundamental knowledge elicitation processes is the act of asking a question. It has to be such that new knowledge emerges in the respondent's answer. Groenendijk [2] said that the semantics of interrogatives still remains an undeveloped part of natural language semantics. He developed a pragmatic view on questions whether his counterpart, Ginzburg, treats questions from a semantic perspective [3]. In the context of current work, *interrogative semantics* is very suitable because it considers the "aboutness" of a question and how a question can be "resolved" by a potential answer [4]. We formally define the QAF triad in terms of Ginzburgean *interrogative semantics* in Section 4. But before that, we shortly introduce the situation theory in Section 2 because interrogative semantics introduced in Section 3 is built solely on situation theory.

In Section 5 we provide a pragmatic description on how knowledge acquisition process can take place with previous requirements fulfilled. The underlying computational model is described in Section 6. In Section 7 we conclude and propose future work.

2. SITUATION THEORY
The underlying idea of the *situation theory* (see Barwise and Cooper [5] [6]) is that the information can be divided into distinct units called *state-of-affairs* (SOAs), also referred as *infons*. These units have the purpose of describing diverse *situations* (*"parts of the world that serve as context"* [3]). It is assumed that SOA's are partially ordered by information subsumption operator[1]. A situation is described by various relations that hold among different objects. The objects (in the world) are considered as invariants that have properties and stand in relations.

[1] Information subsumption operator '→' is defined in Heyting's Algebra named after Arend Heyting (9 May 1898 – 9 July 1980)

Companion Proceedings of *IHM'11, 23ème Conférence francophone sur les Interactions Homme-Machine*, October 24-27, 2011, Sophia Antipolis,France.

Def. (infon [7]***):*** An infon is a structure $\langle R, a_1, a_2 \ldots a_n; i \rangle$ that represents the information that the relationship R holds (if *i=1*) or does not hold (if *i=0*) between the objects $a_1, a_2 \ldots a_n$.

Ginzburg extends the previous definition by saying that basic infons σ are constructed in pairs corresponding to whether the objects assigned to argument roles stand in relationship *R* or do not:

$$\sigma = \langle R, a_1, a_2 \ldots a_n; + \rangle$$
$$\bar{\sigma} = \langle R, a_1, a_2 \ldots a_n; + \rangle$$

Def. (proposition [8]***):*** A proposition is a pair of a situation *s* and an infon σ denoted as **(*s!σ*)**

Def. (fact [8]***):*** A proposition ($s!\sigma$) is TRUE if situation *s* factually supports the infon σ denoted as $\boldsymbol{s \vDash \sigma}$. In such case the infon σ is called a fact.

Def. (abstraction operator $\curlywedge$ [6]***):*** Is a binary operator whose first argument is any indexed family of parameters *F* and whose second argument is any parametric object *o*, denoted as $\curlywedge \boldsymbol{Fo}$.

Note that in this work we will apply abstraction operator only to infons, hence the object *o*, in our case, can be interpreted as an infon *σ*.

Def. (n-ary infon abstract μ [6]***):*** an abstraction of an infon *σ* is the result of abstraction operator on infon *σ* denoted as:

$$\mu = \curlywedge x_1, x_2 \ldots x_n \, \sigma(x_1, x_2 \ldots x_n)$$

Where $x_1, x_2 \ldots x_n$ are the parameters of infon *σ,* i.e., the objects $a_1, a_2 \ldots a_n$, between which its relationship holds.

Def. (instantiation [8]***):*** Is the inverse of abstraction. If there are objects in the world such that they match the parameters of an infon abstract, applying those objects as parameters of the infon abstract is called instantiation. It is denoted as:

$\mathbf{INST}(\mu) = \{\boldsymbol{\sigma} | \exists \mathbf{a_1} \sim \mathbf{x_1}, \mathbf{a_2} \sim \mathbf{x_2}, \mathbf{a_3} \sim \mathbf{x_3} \ldots \mathbf{a_n} \sim \mathbf{x_n} | \boldsymbol{\sigma} = \mu(\mathbf{a_1}, \mathbf{a_2} \ldots \mathbf{a_n})\}$, where $a_i \sim x_i$ means that object *a* is compatible and can take place of parameter *x*.

Next, we introduce the formal interrogative semantics as it is developed by Ginzburg [8] [9] [3].

3. INTERROGATIVE SEMANTICS

The situation theory can be extended to include the notion of question. To do so, Ginzburg proposes a metaphor of *agent engaged in inquiry action.* Taking the assumption that the agent has a set of snapshots of the world[2], some complete, some incomplete or fuzzy that tend to characterize a situation s_0. *Posing a question* means associating the incomplete snapshot i_0 with situation s_0. *Answering the question* means to find another snapshot that completes the missing parts of i_0 and *predicating* that it depicts situation s_0. The question defined by associating i_0 with s_0 ($s_0?i_0$) is *(resolved or) decided* if a snapshot *a* can be found that fills the missing parts of i_0 such that they describe the situation s_0 in the most appropriate way for the agent. *"The notion of aboutness that naturally emerges from this metaphor is one based on informational subsumption, whereas resolvedness is closely related to factuality"* [3]. This view binds the questions and the proposition to the factual state-of-affairs.

Def. (question [8]***):*** Question is a structure that is constructed as a pair of a situation *s* and *a n-ary infon abstract μ* denoted *q*=(*s?μ*), where *s* serves as a pivot to provide connection to the world.

Ginzburg emphasizes that relations definable semantically by a question should include: *aboutness*, *the exhaustive answer* (see the definition of Hamblin answer), and *facts that decide* (or potentially resolves) *the question*.

Def. (Hamblin answer [9]***):*** proposition **p=(*s!τ*)** is an answer to question *q*=(*s?μ*) if *p* is an instantiation of *q* or negation of such instantiation.

- In case of polar question the exhaustive set of Hamblin answers is:
$$\{r | HamblinAns(r, \curlywedge p)\} = \{\mathrm{p}, \neg \mathrm{p}\}$$
- In case of unary wh-question the exhaustive set of Hamblin answers is:
$$\{r | HamblinAns(\mathrm{r}, \curlywedge \mathrm{A}\ \mathrm{p(A)})\} = \{p(a_1), \neg p(a_1), \ldots, p(a_n), \neg p(a_n)\}$$ where A={a1, a2, a3 … an}

Def. (aboutness [8]***):*** *p* is about *q* iff *p* entails a finite disjunction of Hamblin answers.

Aboutness is a relation that, intuitively, captures the range of information associated with a question *q* independently of factuality or level of detail. We say that an infon *τ* is about an abstract *μ* iff *τ* represents an instantiation of μ or negation of such instantiation.

Def. (decidedness [8]***):*** a question **q=(*s?μ*)** is decided by infon τ iff:

- s factually supports infon τ, i.e., $s \vDash \tau$, and either of the two above
- τ is an instance of μ, and s factually supports τ, i.e., $\tau \in$INST(μ) and $s \vDash \tau$
- or there exist a infon σ that is an instance of μ and s factually supports σ such that τ is a negation of σ, i.e., $\exists \sigma | \sigma \in INST(\mu)$, $\tau = \bar{\sigma}$ and $s \vDash \tau$

But why do we need all these definitions? The definitions that have been introduced in section 2 and 3 are the foundation for formal definition of the QAF.

4. FORMAL DEFINITION OF QAF

The triad question, answer, and fact are defined in accordance with Ginzburgean theory of question described in Section 3

Def. (triad-question): a unary question $q = (s\ ? \curlywedge x\ \mu(x))$ where the unknown information *x* is a parameter of the infon abstract *μ*.

Def. (triad-answer): a simple triad-answer is an object denotation *α*, such that the object *α* is compatible with parameter *x* and can be used for instantiation of a triad-question *q*, *x∼α*;

Note that by object denotation we mean any lexical term used to reference an entity from the universe of discourse.

Def. (triad-fact): is a Hamblin answer *f* to the triad–question *q* with a triad-answer, i.e., *f=(s!φ)*, where *φ=INST(μ(α))* and $s \vDash \varphi$

Def. (grounding question): a polar question $w = (s? \curlywedge \varphi)$ that assesses whether a situation *s* factually supports infon *φ*, i.e., $(s \vDash \varphi)$.

For example the question "*Can you give me examples of a president?*" can receive an answer like "*Jacques Chirac*", hence the factual proposition is "*Jacques Chirac is an example of president*". The grounding (factuality assessment) question to be addressed to the respondent is "*Is it true that Jacques Chirac is a president?*"

5. INTERACTION MODEL

The interaction is explained in terms of operations that dialogue actors execute. But before that we first define the following two operations: acceptance of an answer and grounding of a fact.

[2] Wittgenstein picture theory in *Tractatus Logico-Philosophicus* 1922

Def. (acceptance): is when the dialogue participant believes that the proposition *p* exhaustively characterizes the semantics of the utterance *u* produced by another participant.

Def. (answer acceptance): is acceptance of a proposition *a*, that represents an answer to a previously stated question *q*.

Def. (naïve answer acceptance): is a weak answer acceptance where the proposition *a* is assumed to be an answer to question *q*.

Def. (grounding [8]***):*** the epistemic operation through which the dialogue contributor and the rest of the dialogue partners reached the state of mutually believing that everybody understood what the contributor meant to an extent satisfactory for the current purpose of discussion.

Def. (reflexive grounding): is when the dialogue contributor is accepting a statement S_{n+1} as being a correct paraphrase of his previous statement S_n.

A weak implication of reflexive grounding definition is that the contributor might believe that the interlocutor who produced S_{n+1} understood the previous statement S_n.

In a human-to-human interaction, the human interviewer decides the correctness of the answer, by inference on the personal set of beliefs. If the answer "appears" satisfactory then it is grounded as common belief, otherwise the interviewer disambiguates the answer by digging deeper for the meaning of the answer or for another answer to the initial question. The machine-interviewer is truth neutral and consequently naively accepts the answers (this eliminates critical judgment, even common sense).

For the truth checking of the facts we propose to apply the grounding operation reflexively. The hypothetic fact formulated from assumed decidedness of the question by the answer is used as the subject of a polar question that inquires for the truth value of the fact. The respondent, hence, is faced with its own paraphrased answer on which he/she has to decide whether it is correct or not. The answer to the previous polar question serves as grounding criteria. If the respondent agrees on the fact then it is grounded as common belief, else it is rejected.

The pseudo-interaction scenario between a user *U* and a machine *M* is represented in the pseudo code below:

```
//elicitation protocol
M: ask q1
U: answer a1
M: execute:
        - naively accept a1
        - assume that a1 decides q1
        - instantiate proposition f1 as the effect of
receiving the answer a1 to the question q1
        - put f1 into pool of assumed facts
M: ask next questions …

//verification and grounding protocol
M: ask q2(f1)
U: answer a2
M: execute:
    - if a2 true then ground(f1)
    - else disambiguate(q1,a1)
```

Note that `ground(f1)` means to put *f1* into commonly believed facts and remove it from the pool of assumed facts; and `disambiguate(q1,a1)` means to insert into the dialogue plan a disambiguation protocol that depends of the aboutness of the question *q1*. This protocol searches to resolve possible known misunderstandings (polysemy, homograph, metonymy, metaphor) or errors that can occur when answering *q1* type of question.

The following example illustrates a fragment of the dialogue between user and *dionysus-talker*[3] dialogue system about "food":

Talker: What kinds of vegetables do you know?
User: cucumber, tomato, tomatoes, potatoes, turnip
Talker: Can you think of something that vegetables is a kind of?
User: plant,
Talker: If vegetables has parts or components, what are those parts or components?
User: skin, seed, pulp,
Talker: Can you think of something that vegetables is a part of?
User: plant
Talker: What kinds of meat do you know?
User: beef, pork, chicken, lamb,
Talker: Can you think of something that meat is a kind of?
User: nutrients,
Talker: If meat has parts or components, what are those parts or components?
User: *(blank)*
Talker: Can you think of something that meat is a part of?
User: animal, dish, cooking receipt

6. COMPUTATIONAL MODEL OF QAF

Asking the question and receiving the answer are elicitation operations. The fact resulting from merging the question and answer is the actual knowledge that can be formalized and encoded into a knowledge base. To do this transformation, we propose to have predefined interpretation for QAF facts. It is feasible because the aboutness predicate of the questions delimits a class of facts. It becomes feasible to map a natural language expression of the fact to it's a formal counterpart in RDF(S), OWL, KIF, etc. The only problem that remains is the answer processing and understanding. But if we restrict the answer form from natural language to LOWYN grammar then it becomes possible to do the mappings.

LOWYN is an acronym that stands for "List Of Words or Yes/No". This language allows two types of constructs: the (1) agreement constants and (2) the enumeration. We call agreement constants the words that express affirmation and negation in a given language (e.g., in English they are "Yes" and "No"). By enumeration we mean a comma separated list of words. This grammar enforces the answers to be concise and precise information-supplying statements.

Note that the description of LOWYN grammar is compatible and extends the formal definition of triad-answer given in Section 4. The pragmatic extension consists in acceptance of multiple objects α_i that can be substituted to parameter x, i.e. $x{\sim}\alpha_i$. Instead of one object denotation α we accept as answer a set of object denotations A=$\{\alpha_1, \alpha_2 \dots \alpha_n\}$, i.e., the list of words in LOWYN sense.

If QAF triads are stripped from their subjects/objects then we obtain scalable and reusable QAF templates shaped by the aboutness predicate of the question.

Def. (QAF-triad-template):

$$TT = < \text{“}Q\_NL\ template\text{”}, predicate(?x), aType, FT >$$

where TT is a 5-tuple consisting of (1) question natural language template, (2) question aboutness predicate, (3) the answer type according to the nature of the question (y/n answer type for polar questions or l.o.w. answer type for wh-question), (4) the fact template.

Def (QAF-triad-fact-template):

[3] http://code.google.com/p/dionysus-talker/ is a plan-based dialogue manager written in python

$$\boldsymbol{FT} = < \text{“}F\_NL\ template\text{”}, \text{“}F\_FL\ template\text{”}\ >$$

where FT is a binary tuple consisting of (1) the fact natural language template and (2) the fact formal language template.

The set of all the placeholders in the QAF template, beside answer placeholders - a_i, are called *template parameters*.

By natural and formal language templates we mean a string that contains one or more named placeholders which are meant to be substituted with context-dependent lexical terms[4].

QAF triad template example:

$$TT1\ = <\ \text{“}Can\ you\ give\ me\ examples\ of\ a\ \{\$qph\}?\text{”}, instanceOf(qph), low, < \text{“}\{\$a_i\}\ is\ an\ example\ of\ \{\$qph\}.\text{”}, \text{“}:\{\$a_i\}\ rdf\!:\!type\ :\{\$qph\}.\text{”} >>$$

where the template parameters are $\{qph\}$ and answer parameter is a_i. For the fact formal expression we chose RDF/S language in Turtle notation[5]. Note that the aboutness predicate *instanceOf(qph)* is not the same as the predicate in the fact formal expression because the instantiation in RDF language is denoted by the predicate rdf:type.

The elicitation technique is described as a dialogue plan. The dialogue plan consists of triad templates and flow-control operators [1]. In a dialogue interaction, QAF triads are meant to be instantiated and executed contextually. It means that the QAF triad is created as the instantiation of a QAF template with context dependent information. The execution of the previously planned dialogue actions creates a particular dialogue context.

The QAF triad instantiation is evolutional with its execution because instantiation of the fact depends on the answer that is obtained from the user. So it happens in two steps, first the question is instantiated from the template and is executed as a dialogue act, then, after receiving the answer, the fact can be instantiated.

Def. (QAF-triad):

$$T\ = <\ \text{“}QNL\ text\text{”}, predicate(?x), \{a_1, \dots, a_n\}, \{ < \text{“}FNL\ text_1\text{”}, \text{“}FFL\ text_1\text{”} >, \dots , < \text{“}FNL\ text_n\text{”}, \text{“}FFL\ text_n\text{”} >\} >$$

where *T* is a tuple resulting from the process of question template instantiation, answer injection, and fact instantiation. The tuple consists of (1) question natural language expression, (2) aboutness predicate, (3) answer set, (4) a set of fact instances.

Instantiation of the QAF triad is evolutional to the execution of dialogue plan. The dialogue plan prescribes when triads are instantiated and how they are parameterized. In turn, dialogue plan, functionally describes the elicitation techniques [4].

QAF triad example:

$$T1\ = <\ \text{“}Can\ you\ give\ me\ examples\ of\ a\ president?\text{”}, instanceOf(president), \{Jacques\ Chirac, Bill\ Clinton\}, \{<\ \text{“}Jacques\ Chirac\ is\ an\ example\ of\ president.\text{”}, \text{“}:Jacques\_Chirac\ \ rdf\!:\!type : president\ .\text{”} > <\ \text{“}Bill\ Clinton\ is\ an\ example\ of\ president.\text{”}, \text{“}:Bill\_Clinton\ rdf\!:\!type : president\ .\text{”} >\} >$$

[4] PEP-292 (http://www.python.org/dev/peps/pep-0292/)

[5] http://www.w3.org/TeamSubmission/turtle/

7. CONCLUSIONS AND FUTURE WORK

This paper proposes a tool for translating informal to formal knowledge in the context of a human-machine interview in a dialogue form. The informal knowledge is the natural language expression of a question, the answer and the following fact which is the imperative rephrase of the question and answer. The conceptualized knowledge is the formal expression of that imperative rephrase as a fact. The QAF triad acts as a translation mechanism where “fact” is the translation pivot. It has been implemented in a plan-based dialogue system called *dionysus-talker*.

The simplicity of QAF Triads and the LOWYN syntax allows a rigorous semantic control of the interaction and at the same time to preserve the meaningfulness of the dialogue. However, the interaction is not formally covered yet. For the consistency reasons, it is advisable to describe interaction semantics in terms of situation theory. It allows procedural descriptions and is able to capture the semantics of actions.

QAF in its current form allows the incremental construction of knowledge. The major drawback is that the revision of the previously elicited knowledge is minimal or even impossible. We plan to extend the QAF triad concept such that the formal expression of the fact captures the operational impact on the existing knowledge base, i.e., either the current knowledge base is appended with new factual knowledge or the factual knowledge will be modified or deleted in the knowledge base. This is opening up space for frameworks and theories of belief revision[6].

The LOWYN syntax is constraining the form of the answer. Compared to natural language it appears as totally inefficient. So there is a big space for improvement for the answer forms to allow richer answers. The answer syntax can, for example, be adapted to allow expressions in Rabbit language [10], which is much closer to natural language expressions.

8. BIBLIOGRAPHY

[1] Gordon Rugg, Peter McGeorge, and Neil Maiden, "Method fragments," Expert Systems, vol. 17, pp. 248-257, 2000.

[2] Jeroen Groenendijk and Martin Stokhof, "Questions," in Handbook of Logic and Language, J. van Benthem, Ed.: Elsevier/MIT Pess, 1997, pp. 1055-1124.

[3] Jonathan Ginzburg, "Resolving Questions," Linguistics and Philosophy, vol. 18, no. 5, pp. 459-527, 1993.

[4] Eugeniu Costetchi, "Towards Automated Ontology Elicitation Dialogues," University of Luxembourg, Luxembourg, Master Thesis 2010.

[5] Jon Barwise, The Situation in Logic.: Center for the Study of Language and Information, 1989.

[6] Jon Barwise and Robin Cooper, "Simple Situation Theory and its Graphical Representation," , 1991, pp. 38-74.

[7] T. Huibers and P. Bruza, "Situations: A general framework for studying Information Retrieval," , 1994, pp. 3-25.

[8] Jonathan Ginzburg, "Dynamics and the Semantics of Dialogue," in Language, Logic and Computation (Volume 1 - CSLI Lecture Notes), J. Seligman, Ed. Stanford: CSLI, 1996, vol. 1.

[9] Jonathan Ginzburg, "Questions and Internalizing Relevan," in Workshop in Philosophy and Linguistics, University of Michigan, 2009.

[10] Martina Johnson and Catherine Dolbear Glen Hart, "Rabbit: Developing a Control Natural Language for Authoring Ontologies ," Lecture Notes in Computer Science, vol. 5021, pp. 348-360, 2008.

[6] As proposed by Alchourrón, Gärdenfors, and Makinson.