The Imitation Game Revisited - Building Turing's Oracle Machine

Dirk Scheuring
Imaginary Machines
Venloer Strasse 63
50672 Cologne, Germany
+49 221 492 446 4
scheuring@gmail.com

ABSTRACT

Is it possible for human storytellers to create the kind of interactive virtual character that would be needed to successfully play Turing's Original Imitation Game [18]? The implementation of a framework for system of Oracle machines using Turing-recognizable languages that cover a finite range of Turing degrees is shown. It is then demonstrated how this framework can be used to implement an OODA Loop that a virtual character could use as its basic cognitive process. Starting with this basic functionality, I sketch a concrete virtual character and show how it can be created so that, in conversation with Turing's interrogator, it can reflect on the motivations behind all of its actions. I also highlight the crucial role which the storyteller plays in the process of AI creation.

Categories and Subject Descriptors

F.1.1 [**Computation By Abstract Devices**] Models of Computation - *Bounded-action devices (Turing machines, Oracle machines), Computability theory.*

General Terms

Algorithms, Design, Experimentation, Human Factors, Languages, Theory.

Keywords

Interactivity, Actor, Player, Game, Identification, Character, Theme, Plot, Genre, Logic, Story.

1.INTRODUCTION

Alan Turing's Original Imitation Game [20] pitches a man pretending to be a woman (actor A) and an actual woman (actor B) against an interrogator (actor C), who is in another room and cannot see them. Can C, by exchanging written messages with A and B through some mechanism (Turing recommended a teletype, but Instant Messenger seems to be a valid modern-day replacement), find out which of the two is a real woman, and which one is a fake?

Then Turing changes the game's setup: "We now ask the question, ‘What will happen when a machine takes the part of A in this game?’ Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?"

1.1Identification

For a male actor pretending to be female, it is necessary for him to identify with a woman. The WordNet definition that fits this sense of "identification" is "attribution to yourself (consciously or unconsciously) of the characteristics of another person (or group of persons)". In other words: the male actor, A, "plays a role" - that of a woman -, and has to create a character, whose characteristics - female ones, in this case - he has to attribute to himself. If he does this in a believable way, chances are that the interrogator, C, will identify A, through the character he plays, with a woman. This alludes to another sense of "identification", one that WordNet describes as "evidence of identity; something that identifies a person or thing".

It follows that a machine that is assigned to take A's place in playing the role of a woman has to believably project a female character. From this follows that it would have to identify with that female character, in order to stand a chance to be mis-identified by C. And from this follows that that the machine would have to first create that character in order to identify with it, and play its designated role.

1.2The problem

As of now, machines cannot create the kind of characters that they could identify with. Bringsjord and Ferruci [4] extensively researched this topic by building and documenting BRUTUS, a storytelling machine. Bringsjord [5] comes to the conclusion:

> *... fact is, highly creative behavior, whether it be the production of belletristic narrative or the discovery of a startling theorem, is currently inexplicable from the standpoint of computation-based cognitive psychology---and part of the proof of this is the complete absence of any computational artifacts that accomplish the tasks in question, despite AI's concerted effort to create them.*

Elsewhere [6], Bringsjord explains:

> *Renowned human storytellers understand this concept. For example, playwright Henrik Ibsen said: "I have to have the character in mind through and through, I must penetrate into the last wrinkle of his soul." Such a modus operandi is forever closed off to a machine.*

(For a longer version of the Ibsen quote, see [12])

Programming is all about abstraction. At least for developing large progams, some form of abstraction is essential to making computing intellectually manageable. Character is all about individuality, and individuality is the opposite of abstraction.

1.3The question

Is it possible for human storytellers to create such interactive virtual characters as they would be needed to successfully play The Imitation Game, by identifying with them - or, alternatively, by identifying with machines that would be able to create thoses characters -, so that all that would be required of a machine to "play" such a character would be the machine's ability to simply "act it out"?

The meaning of the phrase "acting it out", here, is taken to be equivalent to the meaning of the phrase "doing something that is Turing-decidable".

2.DECIDABILITY

2.1Complete decidability

We say that a formal language is decidable (or recursive, or Turing-computable) if there is an algorithm (such as a Turing Machine, or TM) which always halts after a finite number of steps, answering the question of whether the input string is valid in the language with YES or NO. The number of steps, though always finite for each input, need not be bounded in any specific way related to the size of the input.

2.2Partial decidability

A formal language is said to be partly decidable (or recursively enumerable, or Turing-recognizable) if there is an algortithm (again, instantiated as a TM) that always halts with YES if and only if the input string is valid in the language. If the input string is not valid in the language the algorithm/Turing machine with either halt with NO, or will never halt.

2.3Degrees of undecidability

Between two languages that were both Turing-recognizable, Turing defined a relationship that involved OraclemMachines [19]. That means that, if language A is decidable by a generalized Turing Machine that has "oracular" knowledge of language B, language A is Turing-reducible to language B. The Turing Machine has an instruction that can decide in one step if an input is or is not valid in language B. This feature of the TM is called an "oracle".

2.4The language of the Halting Problem

Turing showed that the halting problem is in effect a partially decidable language: we input the Turing program and wait for the TM to halt. If it does, then the answer is YES. If it does not, we will never get an answer - we will be waiting forever.

This "language" of the halting problem turns out to be Turing-complete: every Turing-recognizable language is reducible to the halting problem "language".

2.5Post's Problem

In 1944 [16], Emil Post asked whether it is possible to have an enumerable set which is neither computable, nor as difficult as the halting problem. The question was answered in the 1950s by Friedberg [9] and Muchnik [14], who, independent of one another, proved that there exist undecidable, effectively enumerable sets that have different *degrees of undecidability*.

The whole of this work, of course, builds on Kurt Gödel's Incompleteness Theorems [10].

3.A GENERALIZED ORACLE - THE CORE CODE

Implementing the framework for a system of Turing-recognizable languages that is structured by a hierarchy of Turing degrees, using a Turing-complete computer language, is a remarkably straightforward process. For our example, we will use the Artificial Intelligence Markup Language (AIML) [7]. AIML is a tail-recursive functional language [23] which is optimized for the semantic processing of strings, where input and output strings are desired to be at the semantic level of sentences in natural language.

3.1The listing

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<aiml version="1.0">

<category>
<pattern>*</pattern>
<template>
"Degree 0: No.
<srai>TEST SENTENCE DEGREE1</srai>
</template>
</category>

<category>
<pattern>* DEGREE1</pattern>
<template>
Degree 1: No.
<srai>TEST SENTENCE DEGREE2</srai>
</template>
</category>

<category>
<pattern>* DEGREE2</pattern>
<template>
Degree 2: No.
<srai>TEST SENTENCE DEGREE3</srai>
</template>
</category>

<category>
<pattern>* DEGREE3</pattern>
<template>
Degree 3: No.
<srai>TEST SENTENCE DEGREE4</srai>
</template>
</category>

<category>
<pattern>* DEGREE4</pattern>
<template>
Degree 4: No.
<srai>TEST SENTENCE DEGREE5</srai>
</template>
</category>

<category>
<pattern>* DEGREE5</pattern>
<template>
Degree 5: No.
<srai>TEST SENTENCE DEGREE6</srai>
</template>
</category>

<category>
<pattern>* DEGREE6</pattern>
<template>
Degree 6: No.
<srai>TEST SENTENCE DEGREE7</srai>
```

```
</template>
</category>

<category>
<pattern>* DEGREE7</pattern>
<template>
Degree 7: No.
<srai>TEST SENTENCE DEGREE8</srai>
</template>
</category>

<category>
<pattern>* DEGREE8</pattern>
<template>
Degree 8: No.
<srai>TEST SENTENCE DEGREE9</srai>
</template>
</category>

<category>
<pattern>* DEGREE9</pattern>
<template>
Degree 9: Yes."
</template>
</category>

</aiml>
```

3.2Discussion

3.2.1Result

Assuming appropriate configuration, the program listed above should run on any implementation of an AIML interpreter that meets the AIML language specification. On any input, one and the same output should be returned:

"Degree 0: No. Degree 1: No. Degree 2: No. Degree 3: No. Degree 4: No. Degree 5: No. Degree 6: No. Degree 7: No. Degree 8: No. Degree 9: Yes."

3.2.2Analysis

This result shows the machine reporting its search through sets of sentences written in languages which, taken together, represent a sequence of successive Turing degrees. Going through the degrees, it says “No” each time it has finished trying a language of a certain degree without finding any but the default match, and then continues its search by going through the language of the next higher degree. After nine “No”s, the recursion halts as a final match is found at degree 9, and the machine says “Yes”.

3.2.3Strategy

Each of the ten AIML categories implements a set of Turing machines whose pattern matcher can match one or more inputs in the language of its own degree, can generate zero or more strings in the language of degree 0 (natural language), and can generate zero or more strings in the language of its own degree, or in a lower-degree language, which are passed on using the `<srai>` (recursion) operator.

3.2.4Knowledge base and thought base

We call each string that can be generated in the language of degree 0 (natural language) part of the knowledge base, and each sentence that can be generated in a higher-degree language, we call part of the thought base.

It is not necessary that the machine reports on each thought in a sequence of thoughts, as our demo application does it. Any string in natural language can also be treated and processed as part of the thought base, provided it is enclosed in AIML's `<think>` tag. However, since we want to demonstrate how a machine's thought can change its direction, we let the machine explicitely enumerate each step.

4.PROGRAMMING AN ORACLE

Let's add one AIML category to our core code:

```
<category>
<pattern>TEST * DEGREE3</pattern>
<template>
Possible match in degree 3 - Halt?
<srai>HELLO PEOPLE DUCKS AND ROBOTS DEGREE3</srai>
</template>
</category>
```

This category matches all input sentences that begin with TEST, end with the degree marker DEGREE3, and have one or more words in the middle that get matched with the wildcard symbol, *.

4.1Issuing an oracle

For our testing convenience, the machine informs us that a match is possible in the set of the degree 3 language, and then it issues an oracle, namely the string `HELLO PEOPLE DUCKS AND ROBOTS DEGREE3` . Which means as much as thinking:, or saying: “Go search for some greeting adressed to people, ducks, and robots in the degree 3 language.”

4.2No such match

Of course, without adding at least one more category to the set, the oracle string doesn't find a match that is better than the pattern:

* DEGREE3

so for any input, the output will now be:

"Degree 0: No. Degree 1: No. Degree 2: No. Possible match in degree 3 - Halt? Degree 3: No. Degree 4: No. Degree 5: No. Degree 6: No. Degree 7: No. Degree 8: No. Degree 9: Yes."

The machine now faithfully reports that it tries for an oracle match, but the degree 3 default ends up getting matched anyway, as gets any successive default, with the machine halting at degree 9, as before.

4.3A partial match

But maybe it's just that a greeting as specific as the one that the oracle specifies is not in the degree 3 language, but theres another, mor generic one: For instance, behind our backs somebody could have added the following category to our code:

```
<category>
<pattern>HELLO * DEGREE3</pattern>
<template>
Yes. As I would  say: 'Hello World!'"
</template>
</category>
```

In which case the output for any input becomes:

"Degree 0: No. Degree 1: No. Degree 2: No. Possible match in degree 3 - Halt? Yes. As I would say: 'Hello World!'"

4.4Stronger Oracle machines

Using this framework, a storyteller could now create categories – as sets of Oracle machines – that, unlike the hardcoded example sentences we have seen, could issue oracles consisting of an arbitrary number of symbols, constants as well as variables, plus the closing degree marker specifying a degree equal to or lower than the degree of the machine which issues the oracle.

4.5Precision vs. abstraction

Starting from natural language – degree 0 – as the *most precise* of the languages that we have at our disposal – it captures the maximum of a charcter's individuality -, the storyteller generates new languages that represent gradual abstractions over the meaning of the sentences of the degree 0 language. Those new languages know their own degree of Turing-undecidability – viz. abstraction -, because they have symbols for that which they know the meaning of. Each language but the one of the highest degree is a partially decidable set, but since the highest-degree language is decidable, the set is decidable as a whole, just like we wanted it to be in 1.3. The definability of the oracle languages can approximate the degree of precision of natural language – degree 0 – to an arbitrary extent, limited only by the resourcefulness of the human storyteller(s).

4.6The general goal

Pertaining to the question of whether the machine in his Imitation Game might, instead of playing the character of a woman, also play any other roles, Turing remarks in [20]:

"It might be urged that when playing the "imitation game" the best strategy for the machine may possibly be something other than imitation of the behaviour of a man. This may be, but I think it is unlikely that there is any great effect of this kind."

We take this quote as confirmation that Turing meant for the machine to be able not only to play the role of a woman opposite to the human who plays the role of the interrogater, but to play any role, and fill that role with any character, that its programmer (s), or storyteller(s), can identify with, and through identification, create. The storyteller employs some non-Turing-computable process which results in the creation of one or more oracles in one or more languages with certain Turing degrees. At runtime, these oracles get inserted into the machine's thought process in a systematic way at the times at which the machine encounters an input that would otherwise be uncomputable. The general goal is to add degrees of computability to uncomputable input.

5.PROGRAMMING AN ACTOR

What else do we know about the Imitation Game? It is an interactive game, for starters; a human interrogator interacts with a virtual actor.

5.1Interactivity

Interactivity is the degree to which in a communication process each message is related to the previous messages exchanged [17]. By this definition, an interactive process begins when actor A sends an arbitrary message X to actor B. B replies by sending back the message Y which refers to X. A then sends the message Z which might refer to X, to Y, to both of them, or to the relationship of all three messages so far. This .process contiues until the inter-actors loose either the motivation or the ability to keep sending messages which refer to other, earlier messages.

5.2Conflict

One thing that we can say of the kind of interactivity that we can associate with the Imitation Game is that we can easily see it as a conflict. The interrogator's goal is to reveal the machine as wearing a mere “character mask”. The machines goal is to convince the interrogator that there is no mask, that it is identic with the character.

5.3Concepts

What else can we know about the generic interrogator? Which concepts is s/he most likely to talk about? Let us look at WordCount.

WordCount [11] is a web application that presents the 86,800 most frequently used English words, ranked in order of commonness. According to this list, the most common words that could be identified with expressions of concepts are “I”, at position 11, and “you”, position 14. No other concepts come even close in commonness. The interrogator can be expected to use the concepts of “I” and “you” a lot, and the virtual character should be expected to be prepared for this.

This is excellent, since it motivates a conflict which lies at the bottom of all possible conflicts between possible characters, and which can be expressed using the rule “I am not you.” Note that this rule can be said to ground the interaction.

5.4More concepts spawning conflicts

WordCount can be “interrogated” also, queried about the rank of a certain word or the word which occupies a certin rank. The use that people make of this feature gets tracked by the QueryCount application. A look at the head of this list shows that words like “sex”, “fuck”, “love”, “shit”, “god”, “penis”, “cunt”, and “ass” are among the most popular queries, and that “I”, at position 7, ranks even higher than in WordCount.

Our goal is to create a Turing-recognizable language which captures conflicts between characters at a level that is more abstract than that of sentences in natural language. It looks as if this preliminary list of commonly used concepts could be a useful starting point, since combinations of these concepts, like “I fuck you” and “you ass”, which seem to be quite probable to occur due to the high ranking of the component words, can also easily interpreted as conflict-related.

5.5Primitive values

Further, the interrogator can be expected to possess of a value system, which makes it possible to assign basic predicates to concepts, like GOOD and BAD, so that basic value propositions can be expressed, such as “Sex is good” and “You are bad”. Also, predicates that signal agreement and disagreement would definitely be needed; we choose the strings POSITIVE and NEGATIVE to serve these roles. .All higher-level behavior is expected to be reducible to this level of primitive discriminations.

It seems obvious that the distribution of associations between concepts and value judgements is particular and unique to any character worth of being called a character. In fact, the very difference between a character and everybody else is what makes a character a character.

5.6Behavioral themes

To define a range of higher-level behaviors that the interrogator can be expected to employ, I use the Five Factor Model of Personality Traits [8], which is frequently used by today's psychologists. The five factors are:

AGREEABLENESS: e.g. "You're right."
CONSCIENTIOUSNES: e.g. "I'm busy."
OPENNESS (to experience): e.g. "Do you like sex?"
EXTRAVERSION: e.g. "Obey my command!"
NEUROSITY: e.g. "Fuck you!"

There are a lot of other models that could be used for this purpose, like Jungian Archetypes [1], the Myers-Briggs model

[3], the PEN model [Eysenck 1991], or the Enneagram model [15]. The Five Factor model was chosen manly because it limits the number of factors to five, which is feasible in terms of the workload on the storyteller, but still provides quite some variety of possible behaviors, from loving to hating.

5.7The “Wh-questions”

Finally, what can be expected from the interrogator is direct interrogation.

The most common form of interrogation has the interrogator asking the so-called "wh-questions": "Who?" - ""What?" - "Why?" - "How?" - Where?" - "When?"

These are the questions that children and scientists ask. These are als the questions that writers like Ibsen ask when they penetrate the last wrinkles of their character's souls.

"Who is the character?"
"What does the character want?"
"Why does the character want it?"
"How does the character try to get what it wants?"
"Where is the character?"
"When is the character?"

5.8Zipf's Law

Another data point that seems to support the choices made so far regarding the most important concepts and predicates comes from one of most well-known interactive characters so far, A.L.I.C.E. [22]. In an essay about *Zipf's Law* [21], A:L.I.C.E.'s creator Richard Wallace published a list of the bot's most frequent atomic input matches. These are the Top 20:

8024 YES
5184 NO
2268 OK
2006 WHY
1145 BYE
1101 HOW OLD ARE YOU
946 HI
934 HOW ARE YOU
846 WHAT
840 HELLO
663 GOOD
645 WHY NOT
584 OH
553 REALLY
544 YOU
531 WHAT IS YOUR NAME
525 COOL
516 I DO NOT KNOW
488 FUCK YOU
486 THANK YOU

(the numbers designate the frequency of a particular input)

Apart from salutations and a few queries about “basic facts” (the character's name, age, and existential orientation), there are two notable groups of inputs in this list:

a. valuations refering to the character's last output (“Yes”, “No”, “I don't know”, etc.)

b. queries refering to the motivation (“Why?”, “Why not?”) and purpose (“What?”) of the character's last output.

5.9Requirements for an actor

At this point I would like to devise of some preliminary requirements that my virtual actor shall have to fulfil:

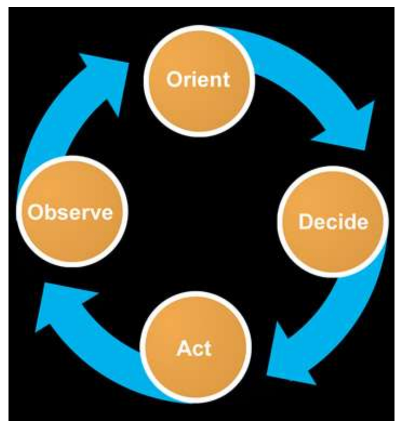
1. Whatever you do or say, always either be able to communicate what motivated your actions, or have a good motivation to conceal your motivations.
2. Every motivation has to be reducible to an *ultimate* motivation – something that the character takes to be a “universal truth”, and which is non-negotiable; e.g. “I am not you.”
3. Considering the existence of this ground conflict, higher-level conflicts are unavoidable. Use them to your benefit.
4. Be consistent – minimize internal conflict. Self-contradiction is only cool if you can show its motivations, and an awareness of the resulting conflict.

6.The OODA Loop

The OODA Loop [2] is a concept originated by military strategist Col. John Boyd (USAF). Its main outline consists of four overlapping and interacting processes:

According to Boyd, decision making occurs in a cycle of observe-orient-decide-act. As the consequences of one's actions get observed, the feedback loop closes.

This, I conjecture, is how the a Turing machine that acts as a character in the Imitation Game should work: make an observation, orientate itself with regards to the context of this observation, decide, act.



I can use my framework of languages of distinct degrees of undecidability to implement the OODA-Loop.

6.1Observation

The process of Observation is handled by the degree 0 language – natural language. There will be a finite set of natural language patterns that I *can* match, and I will try to make the most of them. This means that the input string sent by the interrogator has to be parsed for concepts and predicates, and those which are detected will be stored in working memory. The observation that no observation could be made - the input was unparsable, and matched the wildcard, “*” - is also a valid observation.

6.2Orientation

All observations pertaining to the current input are encoded as an oracle in degree 1 language and thereby passed to the next process, Orientation, which uses the language of Turing degree 1. This language is a bit more abstract than natural language, and is concerned with the arrangement of concepts and predicates that were found in thecurrent input in relation to the context – the results of the interaction so far. The degree 1 language issues another oracle that is readable by oracle machines of degree 2.

6.3Decision

The degree 2 language “talks about” the content of the oracle – concept, predicate, and context – in relation to the character's value system – will the input have a positive or a negative effect on the character's current state. A decision is made – with the decision of not making a decision being a valid decision -, and the Action process is notified via an oracle in the degree 3 language.

6.4Action

The degree 3 language, the highest degree we use, is Turing-decidable by definition, since the actual decision has already been made at a lower degree. This language consists of strings representing abstract actions, with one string, the Null string, representing an explicit non-action, which is considered to be an action, too. The number of possible actions is finite, so that our Oracle machine is guaranteed to halt at this degree.

7.IMPLEMENTATION NOTES

7.1Degree 0

At the Observation level, the degree 0 language deals with input strings in natural language. These strings need to be parsed, deictic expressions need to be resolved, and concepts and predicates have to be disambiguated and stored as they occur.

It turned out that writing a semantic parser in AIML is straightforward. To keep the number of predicates low, I decided to call a concept "well-motivated" if the virtual actor can answer the six "wh-questions" as they relate to that concept.

7.2Degree 1

At the Orientation level, the language is one of tuples of the form (context,predicate,concept). Nested contexts are possible, but it's the storyteller's responsibility to maintain a useful order, and it is possible to introduce an overwhelming complexity here. So for starters, I will limit the possible contexts to the five behaviors accounted for in the Five Factor model.

7.3Degree 2

I need some kind of value system at the Decision level, and again, the Five Factor Model offers at least a useful starting point. Dr. John A. Johnson, Professor of Psychology at Penn State University, offers a web-based implementation of the IPIP-NEO personality test [13], which is based on the Five Factor Model. The test consists of 300 assertion of the form "I get easily irritated", "I trust other people", "I never go on binges", etc., that the testee gets to evaluate on a five-point scale covering the range from "Very unaccurate" to "Very accurate". If, being a storyteller, I now identify with the character I am creating, and take the test on her behalf, answering the questions the way she would answer them, I obtain a useful profile of her personality. As an excerpt:

"Your score on Extraversion is low, indicating you are introverted, reserved, and quiet. You enjoy solitude and solitary activities. Your socializing tends to be restricted to a few close friends.

"Your level of Agreeableness is average, indicating some concern with others' Needs, but, generally, unwillingness to sacrifice yourself for others.

"Your score on Conscientiousness is high. This means you set clear goals and pursue them with determination. People regard you as reliable and hard-working.

"Your score on Neuroticism is low, indicating that you are exceptionally calm, composed and unflappable. You do not react with intense emotions, even to situations that most people would describe as stressful.

"Your score on Openness to Experience is low, indicating you like to think in plain and simple terms. Others describe you as down-to-earth, practical, and conservative."

The actual report is much more detailed, and sure is useful in providing a foundation for the development of a virtual character. This one is probably a bit of a simpleton – a purposeful decision, since I want motivations that limit the size of the needed knowledge base -, but at least it knows how to keep its cool, and it appears to be reliable.

7.4Degree 3

The oracles of the degree 3 level language are strings that describe the action that the virtual character takes as a result of the input in an abstract way, very unlike the natural languge used at degree 0. An Oracle machine that can use that language might look like this:

```
<category>
<pattern>000XPP 00AMODE * XMAIN DEGREE3</pattern>
<template>
<srai>
004OPP
00AEVENT <get name="MAINSPEAK-Event"/>
00AGENRE <get name="genre"/>
MAINSPEAK DEGREE3
</srai>
</template>
</category>
```

8.CONCLUSION

This machine can generate proper oracles, but the average Imitation Game interrogator cannot be expected to be able to understand its output, which is in a language that is at degree 3 of Turing-undecidability.

However, Turing-undecidability does not constitute a problem for a talented storyteller, whose creativity helps her to do things that are technically uncomputable. As the machine halts after creating the final oracle, the storyteller decodes its message and writes a substitute string that translates it up from degree 3 to degree 0. Instead of the oracle text, the interrogator sees a sentence in natural language, written by the storyteller to replace the abstract message of the oracle.

The unfortunate aspect of this creation process for virtual characters is that it is, at its essence, not automatable – its result depends very much on the creativity of the human storyteller. But having read Ibsen and Bringsjord, I had already expected that.

Thefortunate aspect is that the machine at least *aids* the human in the creation process. If the storyteller knows how to read the degree 3 language, knows what those strange-looking strings actually mean that the system generates, she finds all the information she needs to create the character action that is required encoded in the oracle. As needed for an interactive process, successive strings in the degree 3 language relate to their precessors, and their relations prescribe the relations that the equivalent translations into natural language – degree 0 – should incorporate. Provided that the storyteller implements the oracles correctly, all resulting actions should be properly motivated, and the motivations should all be reducible to the ur-conflict: I chose "I am not you", but many others should be possible. The only limit here seems to be the limit of the storyteller's creative ability.

9.REFERENCES

[1] Aurelio, J. M. (1995). Using Jungian archetypes to explore deeper levels of organizational culture. Journal of Management Inquiry, 4, 347-369.

[2] Boyd, J.R. Patterns of Conflict. URL: http://www.d-n-i.net/boyd/pdf/poc.pdf 1986.

[3] Briggs-Myers , I., McCaulley, M.J. , *Manual: A Guide to the Development and Use of the Myers Briggs Type Indicator*. Consulting Psychologists Press (1985).

[4] Bringsjord, S., Ferruci, D.A. Artificial Intelligence and Literary Creativity: Inside the mind of BRUTUS, a storytelling machine. In: *Computational Linguistics,* Dec 2000.

[5] [Bringsjord, S. In Defense of Impenetrable Zombies, Rensselaer Polytechnic Institute, URL: http://www.rpi.edu/~brings/SELPAP/zombies.jcs/zombies.jcs.html , 1996.

[6] Bringsjord, S. Chess Is Too Easy, *MIT Technology Review,* 1998. URL: http://www.technologyreview.com/articles/98/03/bringsjord0398.1.asp.

[7] Bush, N., Wallace, R.S., Artificial Intelligence Markup Language (AIML) Version 1.0.1, 2001, URL: http://www.alicebot.org/TR/2001/WD-aiml/

[8] Digman, J.M., Personality structure: Emergence of the five-factor model. *Annual Review of Psychology, 41,* 417-440 (1990).

[9] [9] Friedberg, R. M. Two recursively enumerable sets of incomparable degrees of unsolvability . *Proc. Nat. Acad. Sci. U.S.A. 43 ,* 236—238, 1957.

[10] Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. In: *I. Monatshefte für Mathematik und Physik,* 38:173-98, 1931.

[11] Harris, J. About WordCount, URL http://www.WordCount.org/about.html, 2003.

[12] Innes, C., *A Sourcebook on Naturalist Theatre*. p 13, London and New York: Routledge 2000.

[13] [Johnson, J.A. The IPIP-NEO (International Personality Item Pool Representation of the NEO PI-R). URL: http://www.personal.psu.edu/faculty/j/5/j5j/IPIP/ , 2000.

[14] [Muchnik, A. A. On the unsolvability of the problem of reducibility in the theory of algorithms. Dokl. Akad. Nauk SSSR (N.S.) 108, 194—197, 1956.

[15] Palmer, H., *The Enneagram:Understanding Yourself and Others In Your Life.* Harper San Francisco (1991).

[16] Post, E. Recursively enumerable sets of positive integers and their decision problems," In: *Bull. Amer. Math. Soc., vol 50* (1944), pp. 284-316.

[17] Rafaeli, S., Interactivity: From new media to communication. In *R. P. Hawkins, J. M. Wiemann, & S. Pingree (Eds.), Sage Annual Review of Communication Research: Advancing Communication Science: Merging Mass and Interpersonal Processes*, 16, 110-134. Beverly Hills: Sage, 1988.

[18] Sterret, S. G. Turing's Two Tests for Intelligence. In: *Minds and Machines Vol. 10 No. 4* pp 541-559, 2000.

[19] Turing, A.M., Systems of Logic Based on Ordinals. In: *Proceedings of the London Mathematical Society, series 2, vol. 45*, pp 161-228, 1939.

[20] [Turing, A. M. Computing Machinery and Intelligence. In: *Mind 49*: 433-460, 1950.

[21] Wallace, R.S., Zipf's Law, *Alice AI Foundation* (2001), URL: http://www.alicebot.org/articles/wallace/zipf.html

[22] Wallace, R.S., The anatomy of A.L.I.C.E., *Alice AI Foundation* (2002), URL: http://www.alicebot.org/anatomy.html

[23] Warren D.H.D., *DAI Research Report 141*, University of Edinburgh 1980.