디지털 캐릭터를 위한 온톨로지 기반의 감성엔진

김지환 조성현 최종학 양정진

가톨릭 대학교 컴퓨터 공학과
정기도 부천시 원미구 역곡 2동 산 43-1
Tel: +82-2-2164-4678, E-mail:{bangga486, ducks5252, izmijjang, jungjin}@catholic.ac.kr

Emotion Engine based on Ontology for Digital Character

JiHwan Kim, SungHyun Cho, JongHak Choi, JungJin Yang

Department of Computer Science ,Catholic University of Korea
43-1 Yeokgok 2-dong, Wonmi-gu, Bucheon, Gyeonggi-do, Republic of Korea
Tel: +82-2-2164-4678, E-mail:{bangga486, ducks5252, izmijjang, jungjin}@catholic.ac.kr

Abstract

디지털 캐릭터가 여러 분야에서 중심적인 역할을 하게 되었고 그에 따라 좀 더 능동적이고 사람에 가까운 캐릭터 개발이 필요하게 되었다. 이러한 요구 중에서 본 논문은 감성기반 캐릭터에 초점을 맞추었고 OCC Model과 AEE Model을 바탕으로 온톨로지와 추론엔진을 이용해서 상황정보를 감정으로 바꾸고 캐릭터의 특성을 반영할 수 있는 Emotion Engine의 Architecture를 제시한 다.

Keywords:

Emotion Engine, Ontology, Inference Engine, OCC model

Introduction

최근 온라인 게임을 비롯하여 영화, 애니메이션 등 가상 공간이 자아 표현과 관계형성 등 " 사회적 상호작용(Social Interaction)"의 공간으로서 각광받고 있다. 대표적인 가상현실 세계인 역할 수행 게임 (Role-Playing Game) 또는 다중접속 역할 수행 게임 (MMORPG : Massively Multi-player Online Role Playing Game)등에서는 인간을 대변하는 캐릭터와 창조된 가공의 캐릭터가 서로 혼합되어 서로 상호작용을 통해서 관계를 형성해 가도록 되어있다. 이러한 역할과 캐릭터 중심의 인간과 인공 구성원이 혼합된 세계의 구성은 인간의 오락과 즐거움을 극대화시키기 위함에 궁극적인 목표가 있다.

이러한 현상은 게임을 비롯하여 교육용 소프트웨어나 모의실험, 모의 시나리오 개발, 영화 애니메이션에서도 주류를 이룬다. 이런 혼합된 세계에서 중요한 요소로 대두되고 있는 부분은 인공 개체간의 상호작용 부분이다. 이것을 위해 인공 캐릭터는 인간과 같은 행동을 보여 줌으로써 인간과 컴퓨터가 구별이 가능하지 않은 수준으로 발전해야 한다.

이러한 인공 캐릭터가 갖추어야 할 것을 3 가지로 요약하면 다음과 같다. 첫 번째, 합리적 지능과 인간과 비슷한 행동을 하는 인공 캐릭터, 두 번째, 환경과 상대 캐릭터의 학습이 가능하고 지능적이고 능동적으로 상호작용하는 캐릭터, 마지막으로 캐릭터의 표현 및 상호작용이 행위 수준뿐 아니라 감성을 기반으로 표현할 수 있는 캐릭터의 필요성이 있다.

본 논문에서는 캐릭터의 표현 및 상호작용이 행위 수준뿐 아니라 감성을 기반으로 표현할 수 있는 캐릭터에 초점을 맞추었다.

우리는 관련연구로 Ortony, Colins, Clore (OCC) Model[1]과 Emotion AI사의 Ian Wilson 의 Artificial Emotion Engine (AEE)[3]에 대해 설명하고

Ontology 개념을 소개한다. 또한 그것을 바탕으로 Emotion Engine Architecture 를 제시하고 지식의 공유, 재사용이 보장되고 논리기반의 추론을 통하여 지식모델의 validity 를 체크하고 지식의 확장을 용이하게 하기 위해서 온톨로지를 사용하였다.

Ortony, Colins, Clore (OCC) 모델

Emotion Engine 을 구성할 때 외부 이벤트를 해석하고 캐릭터에 어떠한 영향을 미치게 될지 판단하는 모듈이 필요하다. 이 모듈을 구성할 때 우리는 기존에 있던 OCC 모델을 도입하여 모듈 구성을 하였다.

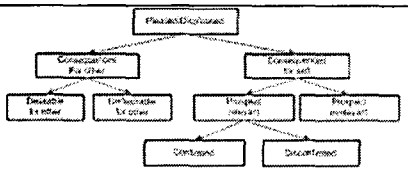
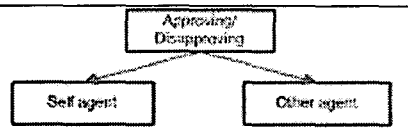
OCC Model[1]은 사람이 표현 가능한 모든 감성을 기술하려고 시도하는 대신, 비슷한 원인에 의한 결과로 생성되어 구별되는 감성을 군집으로 나누고 이것을 감성 유형 이라 정의한다.

감성 평가 요소

OCC Model[1]은 감성 군집을 평가하는 원인을 크게 사건, 개체, 에이전트, 세 가지로 분류한다. 사건은 에이전트의 목표에 관련된 행위를 의미하고 개체는 서로 동등한 자격으로 존재하는 다른 개체들을 의미하고 에이전트는 실제 감성의 주체로 사건과 개체들에 따라 감성 유형을 갖는다.

이러한 세가지 분류 기준 이외에 하위 분류 기준을 적용하여 감성 군집을 다양하게 분류한다.

Table 1. 외부 이벤트 해석 기준 트리

Event 하위의 속성들	
Agent 하위의 속성들	
Object 하위의 속성들	

감성 평가 과정

OCC Model 에서 감성을 평가하는 과정은 Classification, Quantification, Interaction, Mapping, Expression 5 단계로 이루어져 있다.

1) Classification

Classification 단계에서는 캐릭터가 Event 나 Action Object 들에 의해 어떤 Emotional Categories 에 영향을 미치는지 판단한다.

2) Quantification

Quantification 단계에서는 Emotional Categories 에 영향을 미치는 강도를 계산한다.

3) Interaction

Classification 과 Quantification 과정에서 현재의 Event, Action, Object 에 대한 Emotional Value 가 만들어지게 된다. 이 Emotional Value 가 현재 캐릭터의 Emotional Categories 에 영향을 미치게 된다.

4) Mapping

OCC Model 은 22 가지 Emotional Categories 를 정의한다. 이것들은 가능한 최소한의 다른 Emotional Expressions 와 대응되는 것이 필요하다.

5) Expression

Emotional State 는 얼굴 표정으로 직접 나타내어 질 수 있고 캐릭터의 상태에 영향을 줄 수도 있다.

Artificial Emotion Engine (AEE)

Artificial Emotion Engine (AEE)[3]은 Emotion AI 사의 Ian Wilson 이 설계한 Emotion Engine 구조이다. AEE 는 Neuroscience 에서 비교적 명확하게 규명된 Non-cognitive Emotion 을 다룬다. AEE 의 감성 계층은 Personality Psychology 와 Neurology 교수인 Hans Eysenck 와 Jeffery Gray 가 제안한 모델[4,5]을 기반으로 Personality, Mood, Momentary Reaction 3 개의 계층으로 구성된다.

Personality

캐릭터 고유의 특성을 정의하는 부분으로 이것에 의해 외부 이벤트에 의해 캐릭터 감정의 변화 정도가 달라지게 된다. Personality 는 EFA space 에 있는 하나의 좌표 값으로 구성되게 된다.

1) Extroversion

사람의 행동을 구성하는 시스템 중 Approach 시스템에 관련이 있다.

2) Aggression

사람의 행동을 구성하는 시스템 중 flight/fight 시스템과 관련이 있다.

3) Fear

사람의 행동을 구성하는 시스템 중 Behavioral inhibition 시스템과 관련이 있다.

Mood

지속성이 있는 캐릭터의 감정 상태를 저장하고 있는 부분이다. Mood 는 1 차 감정의 집합을 이용하여 결정하게 되는데 감정 집합의 수치 정도는 Personality 에 따라 다르게 된다.

Emotional, Reaction

외부 이벤트에 대한 조건반사적인 감정으로 Mood 에 관계없이 외부로 표현되게 된다. 이때 Reaction 이 외부로 표현되는지의 여부와 표현 강도는 각 캐릭터의 Personality 에 따라 다르게 된다.

AEE 의 Architecture

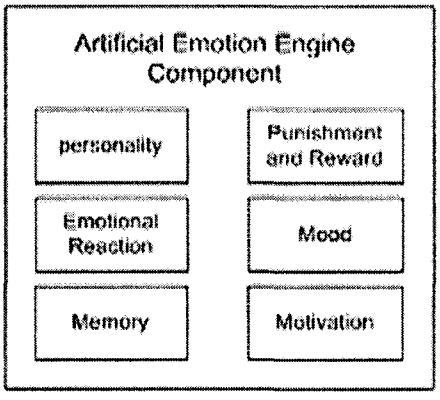


Figure 1. Artificial Emotion Engine 아키텍처

1) Personality 모듈

외부 이벤트가 발생하고 캐릭터의 Mood가 바뀌거나 Emotional Reaction이 생길 때 캐릭터의 특성을 반영하는 모듈이다. Personality 모듈에서 Extroversion, Aggression, Fear 속성을 얻어와서 Mood나 Emotional Reaction에 적용 시킨다.

2) Punishment and Reward 모듈

Punishment and Reward 모듈을 외부 이벤트가 캐릭터의 감정 변화에 좋은 영향을 미칠지 아니면 나쁜 영향을 미칠지 판단을 하는 모듈이다. 여기서는 복잡하게 해석하지 않고 간단하고 확실히 알 수 있는 부분만 해석하게 된다.

3) Emotional Reaction

외부 이벤트에 의해 즉각적으로 일어나는 반응을 만들어 내는 모듈이다. 이 모듈은 캐릭터의 Mood와 상관없이 이벤트 자체만의 Reaction을 생성해 낸다.

4) Mood

캐릭터의 지속적인 감정 속성인 Mood를 변화시키는 모듈이다. Mood를 변화 시킬 때는 Personality 모듈과 Punishment and Reward 모델들과 연동해서 Mood 상태를 변화시키게 된다.

5) Memory

외부 이벤트가 단순하지 않고 복잡한 이벤트일 경우 Punishment and Reward 모듈에서 간단히 해석되지 못할 경우가 생기게 된다. 이럴 경우 Memory 모듈의 도움을 받아 과거의 경험을 살펴보고 이벤트에 대한 판단을 내릴 수 있게 된다.

6) Motivation

같은 이벤트가 발생되더라도 캐릭터의 현재 상태에 따라 감정 변화가 달라질 것이다. 이런 현상을 반영하기 위해 Motivation 모듈의 도움을 받는다. Motivation 모듈은 Maslow의 욕구단계이론[6]을 근거로 하여 만들어 졌으며 캐릭터의 욕구 충족 상태를 저장하고 있다. 그래서 이벤트가 발생했을 때 현재 캐릭터의 욕구 상태를 검사하여 그에 맞는 감정 변화를 일으킬 수 있게 된다.

Ontology

W3C 에서는 데이터를 위한 의미(Meta-data)를 기술하기 위한 표준으로 XML 을 권고한다. XML 은 기계가 읽을 수 있는 구조로 정의 되었고 XML Schema 와 더불어 성분(Element)과 속성(Attribute)으로 메타데이터 어휘를 기술하고 개체 수 제약(Cardinality Restriction)을 이용하여 구문을 기술

한다. RDF[7]는 XML을 기반으로 유, 무형의 자원을 주어 술어 목적어 기반의 트리플 형태로 선언하며 URI 와 더불어 유일한 메타데이터를 기술하기 위한 기반구조로 사용된다. RDF Schema 는 RDF 의 용어, 역할의 개념을 정의하는 경량의 온톨로지를 정의하기 위한 메타언어로 사용되며 응용계층은 각 개념들이 선언된 온톨로지를 명시하여 의미적 상호 운용성을 보장 받는다. OWL[8]은 W3C 에서 권고하는 온톨로지 언어이며 서술논리(Description Logic)를 기술하여 논리적인 표현과 합리적인 시스템을 위한 자료구조로 사용될 수 있다.

온톨로지의 메모리 상주 모델 처리 방식은 방대한 설계 지식을 수용하기에 부적절하기 때문에 온톨로지 저장소의 사용을 요구하고 논리적으로 표현된 지식의 일관성과 추론을 위해 추론엔진의 사용이 요구 된다. 온톨로지 저장소의 기능적 요구사항과 성능적인 요구사항을 충족시키기 위해서 Minerva 등이 연구 되었다.

본 감성 엔진에서는 감정에 관한 정보를 기술하기 위해 OWL 을 이용하여 온톨로지를 제작하고 온톨로지 저장소를 활용하기 위해 Minerva 를 사용한다.

Minerva

실제 온톨로지를 저장하기 위한 메커니즘으로써 온톨로지 저장소의 기능적 요구사항과 성능적인 요구사항을 충족시키기 위해 최근에 IBM 사에서 개발되었다.

Minerva 의 특징

Minerva는 관계형 데이터 베이스로 만들어져 있고 방대한 크기의 온톨로지를 수용할 수 있다. 그리고 W3C의 OWL DL 온톨로지의 저장과 SPARQL 질의 언어를 완전하게 지원하고 TBox 추론과 ABox 추론을 위한 DL Reasoner를 사용한다.

Minerva 의 구조

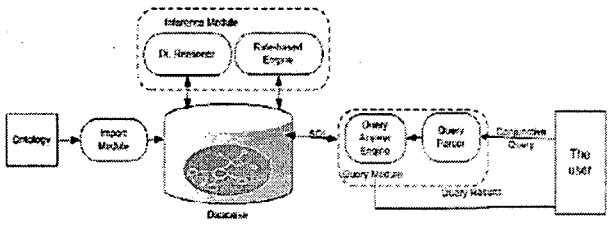


Figure 2. Architecture of Minerva

1) Front-end

사용자로부터의 질의 응답을 하기 위한 Query Module은 SPARQL을 통하여 받아진 질의를 SQL로 변환하여 데이터 베이스로부터 질의를 처리하고 자바 객체 형태로 그 결과를 반환한다.

2) Back-end

온톨로지의 저장과 Inference Module로부터 온톨로지의 추론이 이루어 진다. Inference Module과 데이터 베이스와의 통신은 DIG 인터페이스를 통하여 이루어 진다.

Emotion Engine Architecture

Emotion Engine Architecture 은 Figure 3 에서 볼 수 있듯 크게 Context 해석 모듈과 감성을 혼합하는 Creating Emotion, Personality 와 Mood 그리고 Emotional Reaction 을 반영하는 Character Emotion 3 가지 부분으로 나누어진다.

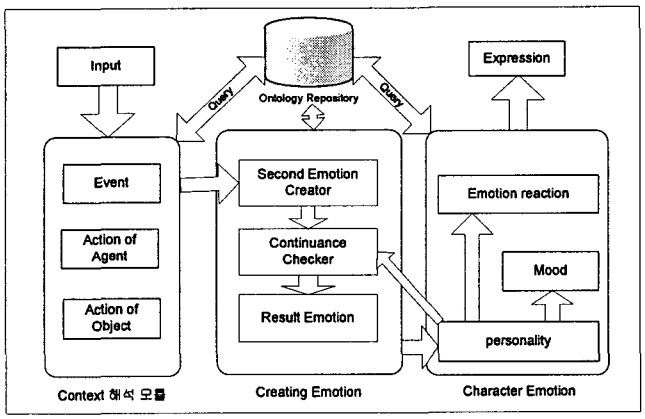


Figure 3. Emotion Engine 의 전체 아키텍처

1) Context 해석 모듈

Context 해석 모듈은 외부에서 들어오는 상황 정보를 OCC모델을 기준으로 분석하여 1차 감성을 도출하는 역할을 한다. OCC 모델의 효율적인 구상과 확장을 위해 온톨로지로 구성하고 Context 해석 모듈은 구 성된 감 성 온톨로 지 저장 소에 질 의를 하여 1차 감성을 얻어낼 수 있다. 감성 온톨로지 저장소에 질의하는 문제는 뒤에 언급하도록 하겠다.

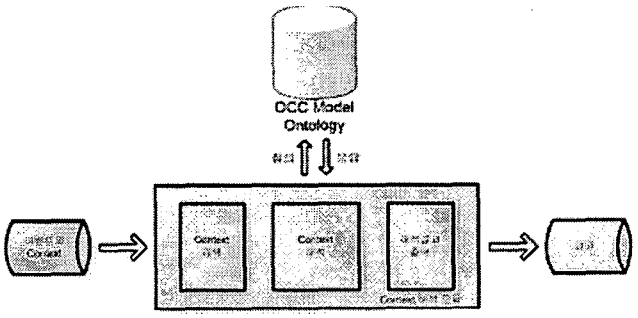


Figure 4. Context 해석 모듈

2) Creating Emotion

Creating Emotion은 Context 해석 모듈에서 나온 감성 결과를 캐릭터의 이전 감성과 연관시켜 경우에 따른 더욱 정교하고 다양한 감성 표현 하는데 목적이 있다. 이는 Second Emotion Creator에서 생성되어 캐릭터의 다양한 감성 표현을 가능하게 한다. 캐릭터의 최근 1차 감성은 Personality의 지속성에 의하여 특정 시간이 지날 경우 소멸시켜 캐릭터의 감성이 중립적인 감성으로 돌아오게 되어 새롭게 표현될 감성에 영향을 미치지 않게 하는 Continuance Checker가 존재 한다.

2차 감성은 Whissel이 작성한 정서 동그라미의 감성이 위치 한 각에 따른 계산에 의해 얻을 수 있다. 예를 들면 Angry는 212°이고 Disappointment는 136.7°이다. 이 둘의 평균 각은 174.35°로 정서 동그라미에서 Indignant를 나타낸다. 이러한 1차 감성들과 2차 감성의 관계를 감성 온톨로지에 저장, 질의를 통해 최종 감성의 종류를 결정하게 된다.

Table 2. 2차 감성 테이블

1차 감정	각도	1차 감정	각도	2차 감정	각도
Angry	212	disappointment	136.7	Indignant	174.35
joy	323.4	remorse	123.3	Hostile	223.35
disappointment	136.7	satisfaction	326.7	Aggressive	231.7
resentment	176.7	satisfaction	326.7	Wondering	251.7

3) Character Emotion

캐릭터의 특징을 나타내는 Character Emotion은 크게 Personality, Mood, Reaction Emotion으로 나뉘게 된다. 감성이 표현되는 상황에 따라 즉각적인 반응을 요구할 경우 Emotion Reaction에서 처리하게 되고 일반적인 반응을 요구할 경우 Mood를 이용하여 처리하게 된다. 캐릭터 별로 표현의 차이를 갖게 하기 위하여 Personality를 참조하게 된다.

Personality는 캐릭터의 특징을 나타내는 것으로 EFA Space를 이용하여 나타낸다. EFA space는 Extroversion, Fear, Aggressive의 각 축에 대한 백분율을 이용하여 나타나게 된다.

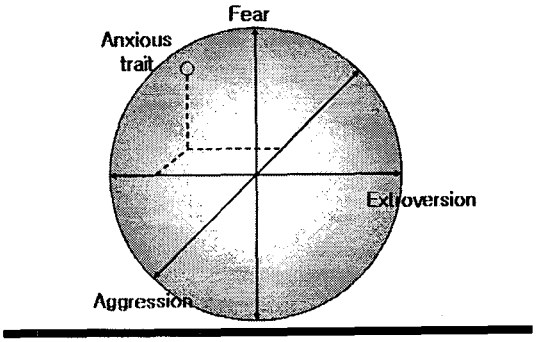


Figure 5. EFA Space

예를 들어 Figure 5에서 Extroversion이 30 이고 Fear가 70, Aggression이 -10이면 Anxious한 특성을 갖게 된다. 이러한 EFA space를 이용하여 캐릭터의 Personality를 구현하기 위해 Figure 6와 같은 모듈을 나타낼 수 있다.

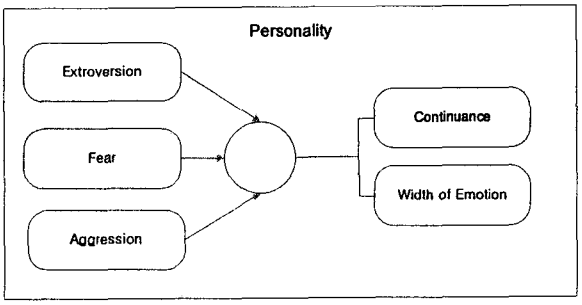


Figure 6. Personality Module

Personality Module의 결과로 감성의 세기와 지속성을 나타내고 이 두 요소 중 지속성은 이전에 언급한 Continuance Checker에 영향을 주어 감성의 지속 시간을 제어한다.

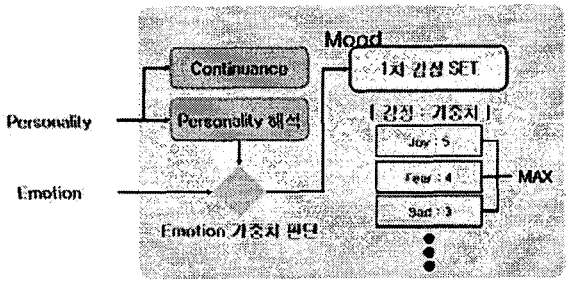


Figure 7. Mood Module

Mood는 캐릭터의 상태를 저장하고 있는 부분으로 1차 감성 집합을 이용하여 개별로 가중치를 주어 캐릭터의 Mood를 결정하게 된다. 이따 각 상황에 따라 반영하는 정도는 Personality의 Width of Emotion에 따라 반영하게 된다.

Emotion Reaction은 즉각적인 반응에 대해

Personality를 이용하여 표현의 크기를 결정하여 바로 표현하게 된다. 이는 Mood와는 관계없이 표현된다.

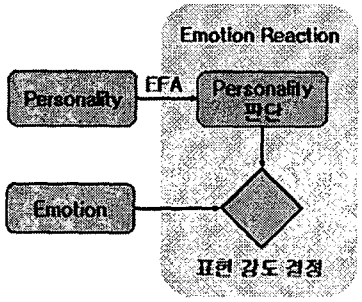


Figure 8. Emotion Reaction Module

온톨로지 표현 및 모델링 기법

Emotion Engine에서 지식의 공유, 재사용이 보장되고 논리기반의 추론을 통하여 지식모델의 validity를 체크하고 지식의 확장을 용이하게 하기 위해서 각 모듈의 내용을 온톨로지로 작성하고 그것을 위해 우리는 Protégé 3.1[11]을 사용하였다.

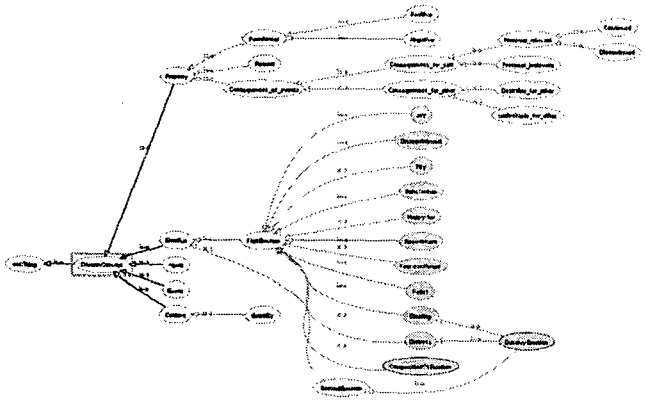


Figure 9. 감성과 감성 판단 온톨로지

Figure 9에서 보여지듯 우리는 1차 감정과 2차 감정 그리고 감정판단 기준에 대한 컨텐츠를 온톨로지로 제작했다.

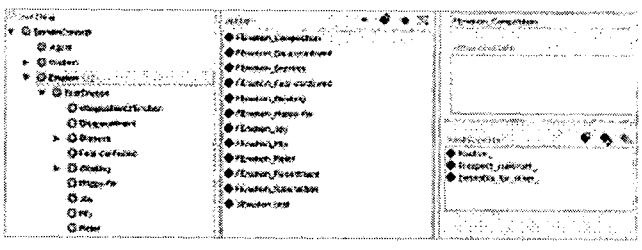


Figure 10. 감성 연계요소

Figure 10은 감정 연계요소를 온톨로지로 나타낸 그림이다. 우리는 Protege3.1를 이용해서 만든 온톨로지의 주요기능은 다음과 같다.

- 감성 컨텐츠를 온톨로지화
- 서술 논리 기반 감성 추론 규칙을 작성
- 응용 계층간 의미적인 상호 운용성을 보장
- 작성된 온톨로지의 재사용
- 감성 컨텐츠의 온톨로지화

온톨로지 저장소와 추론엔진의 활용

온톨로지 저장과 관리의 효율성을 위해 온톨로지 저장소의 사용이 필요하게 되었다. 온톨로지 저장소를 활용하기 위해 Minerva를 사용하여 온톨로지를 저장한다. 저장된 온톨로지에 질의를 통하여 정보를 얻어오게 되는데 질의를 했을 때 추론을 수행하게 되면 시간이 오래 걸리게 된다. 그래서 온톨로지를 저장소에 넣을 때 추론 과정을 먼저 수행해서 넣게 되었는데 이것은 Minerva에서 지원하는 추론기를 사용하였다.

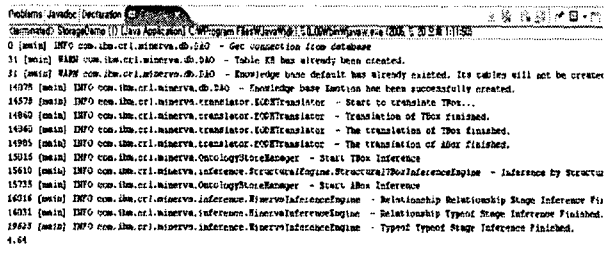
온톨로지 적재 과정

Minerva를 이용하여 OWL 파일을 Relation Data Base(RDB) 형태로 바꾸어서 Database에 저장한다.

```
        minerva.addZOODModel(catofood,false);

        //do inferences
        minerva.commit();
    }

    //load owl documents into ontology repository
    public static void loadOWLDocuments() {
        //generate an ontology store manager
        OntologyStoreManager minerva = new OntologyStoreManager();
        //delete an ontology store
        minerva.deleteOntologyStore("Emotion");
        //new an ontology store
        minerva.newOntologyStore("Emotion");
        //select an ontology store
        minerva.selectOntologyStore("Emotion");
        //load an owl documents into the selected ontology store
        minerva.addOWLDocument("Emotion.owl", false);
        //do inference
        long time = System.currentTimeMillis();
        minerva.commit();


        //
        for (int i = 0; i < 1; i++) {
            minerva.addOWLDocument("file:/c:/owl-ontology/
```

Figure 11. 온톨로지 저장 소스

Minerva가 지원하는 Reasoner가 동작되어 Abox 추론과 Tbox 추론이 실행된 후 RDB형태로 변환되어 DB에 저장된다.



```
Problems  Javadoc  Declaration
<terminated> StorageDemo (1) [Java Application] C:\Program Files\Java\jre1.5.0_06\bin\javaw.exe (2006. 5. 20 오후 1:11:58)
0 [main] INFO com.ibm.cri.minerva.db.DAO  - Get connection from database
31 [main] WARN com.ibm.cri.minerva.db.DAO  - Table KB has already been created.
31 [main] WARN com.ibm.cri.minerva.db.DAO  - Knowledge base default has already existed. Its tables will not be create
14078 [main] INFO com.ibm.cri.minerva.db.DAO  - Knowledge base Emotion has been successfully created.
14579 [main] INFO com.ibm.cri.minerva.translator.KOSTranslator  - Start to translate TBox...
14860 [main] INFO com.ibm.cri.minerva.translator.KOSTranslator  - Translation of TBox finished.
14360 [main] INFO com.ibm.cri.minerva.translator.KOSTranslator  - The translation of TBox finished.
14905 [main] INFO com.ibm.cri.minerva.translator.KOSTranslator  - The translation of ABox finished.
15015 [main] INFO com.ibm.cri.minerva.OntologyStoreManager  - Start TBox Inference
15610 [main] INFO com.ibm.cri.minerva.inference.StructuralEngine.StructuralTBoxInferenceEngine  - Inference by Structu
15735 [main] INFO com.ibm.cri.minerva.OntologyStoreManager  - Start ABox Inference
16016 [main] INFO com.ibm.cri.minerva.inference.MinervaInferenceEngine  - Relationship Relationship Stage Inference Fi
16031 [main] INFO com.ibm.cri.minerva.inference.MinervaInferenceEngine  - Relationship Typeof Stage Inference Finished.
19625 [main] INFO com.ibm.cri.minerva.inference.MinervaInferenceEngine  - Typeof Typeof Stage Inference Finished.
4.64
```

Figure 12. 온톨로지 저장 결과 화면

온톨로지 저장소로의 질의와 응답 과정

SPARQL[10]을 이용하여 온톨로지 저장소로 질의를 하면 그에 적합한 결과를 받을 수 있다 .

```
String sparql=  "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> "+
                "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> "+
                "PREFIX owl: <http://www.w3.org/2002/07/owl#> "+
                "PREFIX : <http://idis.catholic.ac.kr/owl/Emotion.owl#> "+
                "SELECT ?d WHERE "+

                //Emotion의 모든 인스턴스를 i에 바인딩
                "{?i rdf:type :Emotion}" +
                //i에 바인딩된 인스턴스 중 목적어 자격 (목적어가 없음)
                "{?i :hasProperty :Confirmed_}" +
                "{?i :hasProperty :Positive_}" +
                //i변수의 type을 d에 바인딩
                "{?i rdf:type ?d}" +
                //d에 바인딩 된 클래스중 FirstEmotion의 상 클래스 추출
                "{?d rdfs:subClassOf :FirstEmotion}";
```

Figure 13. 감성 온톨로지 질의 코드

```
http://idis.catholic.ac.kr/owl/Emotion.owl#Satisfaction
```

Figure 14. 질의 응답 결과

Results

본 연구를 통해 디지털 캐릭터에 감성을 판단할 수 있는 방안으로 온톨로지 기반의 Emotion Engine을 제시하였다. 구체적으로 온톨로지가 갖는 포괄적인 개념을 언급하면서 구현을 위한 도구와 기술들을 정립하였고 Emotion Engine의 컨텐츠를 온톨로지로 표현하고 서술 논리 기반 감성 추론 규칙과 시스템 아키텍처를 도출하였다.
이로써 기존의 디지털 캐릭터가 무표정하고 피동적인 캐릭터에서 능동적인 캐릭터의 기반을 만들어 주었고 행동중심적인 캐릭터 시장에서 감성을 섬세하게 표현하는 표현중심의 새로운 캐릭터 사장을 개척할 기반을 만들어 주었다.
향후 본 연구에서는 구현된 결과들이 유기적으로 연계되어 보다 자동화에 가깝게 수정이 되어야 하겠다. 또 캐릭터 별 특징에 대한 것들을 좀 더 세밀하게 구현해서 지금보다 다양한 캐릭터를 표현할 수 있도록 해야 하겠다.

Acknowlwdgement

본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 지역문화산업연구센터(CRC)지원사업의 연구결과로 수행되었음

Reference

[1] Ortony .A, Clore. G, and Collins. A, (1988) "The Cognitive Structure of Emotions", Cambridge: Cambridge University Press

[2] Igor S. Pandzic and Rebert Forchheimer, (2002) "MPEG-4 Facial Animation : standard, Implement and Applications", p156

[3] Ian Wilson (2000) "The Artificial Emotion Engine : Driving Emotional Behavior" AAAI

[4] Eysenck H.J. & Eysenck, M.W. (1985) " Personality and individual differences", Plenum

[5] Gray J.A. (1970) "Behaviour research and therapy", Elsevier Science Ltd

[6] Maslow A.H. (1997) "Motivation and Personality", Addison-Wesley

[7] 3C (2004) RDF Primer W3C Recommendation 10 February, http://www.w3.org/TR/2004/REC-rdf-primer-20040210

[8] W3C (2004)OWL Web Ontology Language Overview W3C Recommendation February, http://www.w3.org/TR/2004/RECowl-features-20040210

[9] IBM (2006) IBM Integrated Ontology Development Toolkit March http://www.alphaworks.ibm.com/tech/semanticstk

[10] W3C (2005) SPARQL Query Language for RDF 20 February, http://www.w3.org/TR/rdf-sparql-query

[11] http://protégé.standford.edu