

目次

【국문초록】	xi
I. 序論	1
1. 연구의 배경 및 목적	1
2. 연구의 범위 및 방법	4
3. 논문의 구성	9
II. 이론적 배경 및 선행 연구	11
1. 이론적 배경	11
가. 시맨틱 웹(Semantic Web)과 온톨로지(Ontology)	11
나. RDF(Resource Description Framework)	14
다. RDF Schema	23
라. OWL(Web Ontology Language)	27
마. 시맨틱 웹과 LOD(Linked Open Data)	42
2. 선행 연구 및 사례 분석	44
가. 선행 연구	44
나. 사례 분석	48
3. 도구의 활용	65
가. Protégé	65
나. Altova SemanticWorks	66
다. Validation Parse	67
라. TBC	69
마. WebVOWL	69

III. 역사 자료 연계 및 통합 방안	71
1. 『通鑑節要』과 『資治通鑑』	71
2. 온톨로지 설계 절차	82
가. 목적 및 범위	83
나. 온톨로지 설계	84
3. 클래스 설계	94
가. URI 설계	94
나. 행위의 주체(Agent)	99
다. 행위(Action)	104
라. 지명(Place)	109
마. 왕조(Dynasty)	111
바. 사건(Event)	112
사. 시간(Time)	114
아. 제도(System)	116
자. 어휘	117
차. 기타	122
4. 프로퍼티의 설계	123
가. 개체 간 관계의 표현	124
나. 오브젝트 프로퍼티(ObjectProperty)	131
다. 데이터 타입 프로퍼티(DatatypeProperty)	155
라. 어노테이션 프로퍼티(Annotation Property)	162
5. 기존 온톨로지 재사용 검토	164
IV. 지식 관계망의 설계	167
1. 지식 관계망	167

가. 지식 관계망의 개요	167
나. 시간 관계(hasTime-Span)	168
다. 어휘 관계(termRelation)	175
2. LOD 발행 절차	188
가. LOD 발행 개요	188
나. LOD 구축 과정	199
다. 활용 방안	238
라. 시맨틱 질의를 통한 실험 및 평가	245
V. 결론	258
참고문헌	262
【영문초록】	271

표 목차

표 1. 온톨로지 구성 요소	13
표 2. 온톨로지 노드와 속성 관련 용어	16
표 3. XML 스키마에서 정의된 데이터 형식	20
표 4. 구체화에 필요한 클래스	23
표 5. OWL 언어 비교	28
표 6. RDF-RDFS-OWL Table of Contents	30
표 7. Object Property와 DataType Property 비교	33
표 8. 프로퍼티 제약	36
표 9. ‘문서의 웹’과 ‘데이터의 웹’ 차이점	43
표 10. 사건과 관련한 부가 정보	46
표 11. 국내 LOD 발행 현황	49
표 12. 한국사 LOD 클래스 정의	51
표 13. ObjectProperty	51
표 14. Data type Property	53
표 15. 한국사 LOD의 사건에 사용된 용어의 빈도	57
표 16. The EuroWordNet Top-Ontology	62
표 17. 『資治通鑑』과 『通鑑節要』의 卷次	73
표 18. 『譯註 通鑑節要』의 체재	78
표 19. 『通鑑節要』의 데이터 구조	79
표 20. 『譯註 通鑑節要』의 자료량	79
표 21. 구축 절차	82
표 22. 요구사항 정의서	84
표 23. 정보통신단체표준 개체명 태그세트 및 태깅 말뭉치	90
표 24. 『한국문집총간색인』 등재 용어의 유형 분류	90
표 25. 『譯註 通鑑節要』의 원문과 번역문에 태깅한 개체명 예시	92
표 26. 『通鑑節要』에 나타난 개체명 분포	94
표 27. 본 논문에서 사용한 네임 스페이스	98
표 28. Agent 클래스 분류	99
표 29. 『通鑑節要』에 나타나는 後周 世宗의 표현	100
표 30. 『譯註 通鑑節要』의 본문 구조 예시	101
표 31. 행위를 나타내는 동사의 빈도	108
표 32. 중국에서 건국된 梁나라	112
표 33. 시간 클래스 정의	116

표 34. system	117
표 35. 원문 번역문 대역어 예시	120
표 36. 대역어의 RDF 그래프 표현	121
표 37. 대역어 연계를 위한 클래스	122
표 38. Artifacts	122
표 39. 클래스	123
표 40. 코어넷의 인간 관계	126
표 41. CBDB 인물 관계 예시	127
표 42. CIDOC CRM Class Hierarchy	128
표 43. RELATIONSHIP 온톨로지 어휘	130
표 44. 인물 관계 프로퍼티	133
표 45. hasRelatedSocial	136
표 46. eventTimeRelation	137
표 47. 의미역(Semantic Role) 비교	140
표 48. eventRoleObject	151
표 49. 공간 정보 관련 오브젝트 프로퍼티	152
표 50. 죽음을 나타내는 포뮬어	153
표 51. 죽음을 표현하는 어휘의 예시	153
표 52. 서명 정보	154
표 53. 인과 관계	154
표 54. 사건 관계	155
표 55. 기타 오브젝트 프로퍼티	155
표 56. eventRoleData	156
표 57. date 프로퍼티	158
표 58. 인명 정보(personInformation)	160
표 59. 공간 위치 정보	162
표 60. 관직	162
표 61. 동 시대 복수의 왕조의 예	169
표 62. 동일한 年號를 사용하는 王朝의 예	170
표 63. 중복되는 개체명의 식별 방안	171
표 64. owl:sameAs를 이용하여 서기년(Year)과 연호, 묘호의 연결	172
표 65. 구간을 가지는 시간의 예	173
표 66. 왕조와 관련된 정보의 연결	173
표 67. dynastyStart	174
표 68. hasTime-Span	175
표 69. hasSynonym	176

표 70. hasAntonym	177
표 71. 대역어 관계	177
표 72. BT(NT)	179
표 73. hasKind(isKindOf)	180
표 74. hasBranch(isBranchOf)	181
표 75. hasComponent(isComponentOf)	182
표 76. hasMember(isMemberOf)	182
표 77. containsSubstance(isSubstanceOf)	183
표 78. hasIngredient(isIngredientOf)	183
표 79. spatiallyInclude(spatiallyInside)	184
표 80. hasInstance(isInstanceOf)	186
표 81. hasPart(isPartOf)	186
표 82. SQL 쿼리를 이용한 RDF 트리플 변환 예	194
표 83. RDF 데이터베이스	195
표 84. LOD 발행 도구	196
표 85. 지식 연계 자료량	200
표 86. XML 문서 내에 RDF 트리플의 표현	201
표 87. LOD 구축량	246
표 88. SPARQL의 유형	247
표 89. 기본적인 SPARQL 구문	247
표 90. 주어의 생략 SPARQL	248
표 91. 주어·술어의 생략	248
표 92. 馮廷諤의 신분과 국적을 표현하는 SPARQL	250
표 93. 기본적인 SPARQL 명령어	250
표 94. 유효성 검증을 위한 SPARQL 조회	250
표 95. 五代十國 시대에 황제를 시해한 사람은?	252
표 96. RDF 트리플로 표현한 後梁 太祖 朱全忠의 가족 관계	253
표 97. 朱友貞과 朱友珪의 관계를 조회하는 쿼리	254
표 98. 朱全忠이 蔣玄暉를 시켜 唐昭宗을 시해했음을 조회하는 쿼리	255
표 99. 912년에 발생한 사건을 조회하는 쿼리	256
표 100. 馮廷諤이 어떤 行爲를 했는지 찾는 쿼리	257

그림 목차

그림 1. 공백 노드를 이용한 시간과 공간의 표현	5
그림 2. 시맨틱 웹 아키텍처	13
그림 3. rdf 그래프의 예	14
그림 4. 단순 RDF 트리플 구조	15
그림 5. 공백 노드의 예시	21
그림 6. 공백 노드를 이용한 복수의 속성 표현	21
그림 7. 구체화 그래프	23
그림 8. 'rdf:type'을 이용하여 'rdfs:Class'에 속함을 명시	24
그림 9. 'rdf:type'을 이용하여 rdfs:Property에 속함을 정의	25
그림 10. RDFS와 OWL의 프로퍼티	26
그림 11. RDF와 RDFs의 프로퍼티	27
그림 12. RDF 그래프	34
그림 13. 項羽와 義帝의 관계를 표현한 RDF 그래프	35
그림 14. 李克用의 자손의 RDF 그래프 표현	38
그림 15. 이행성 속성의 예	39
그림 16. 함수 프로퍼티의 사용 예	41
그림 17. 역함수 프로퍼티의 사용 예	42
그림 18. 역사적 사건 지식의 온톨로지 표현	45
그림 19. 기간을 표현한 복문의 예	47
그림 20. 인과 관계 그래프	48
그림 21. 項羽를 이용한 N항 관계의 표현	60
그림 22. Dbpedia와 Wikipedia 비교	63
그림 23. 링크드 오픈 데이터 연결 현황	64
그림 24. Protégé 실행 화면	66
그림 25. Altova SemanticWorks 실행 화면	67
그림 26. W3C에서 제공하는 Validation Parse	68
그림 27. OWL Validator	68
그림 28. TopBraid Composer 실행 화면	69
그림 29. WebVOWL 실행 화면	70
그림 30. 순환 관계의 예	87
그림 31. Schema.org의 Person에 대한 메타데이터	88
그림 32. 행위에 시간과 공간의 추가된 형태	105
그림 33. 공백 노드를 이용한 시간과 공간의 표현	106

그림 34. 의미역을 이용한 다항 관계의 표현	107
그림 35. 행정 구역의 표현 방법	111
그림 36. 사건의 개념	112
그림 37. 시간 간 발생할 수 있는 관계의 유형	137
그림 38. 동명이인의 처리	160
그림 39. 온톨로지 설계 절차	164
그림 40. Swoogle 시맨틱 검색 엔진	165
그림 41. 서기년과 연호의 매핑	172
그림 42. 李滉을 중심으로 한 계층 관계의 표현	178
그림 43. 공간적 부분 관계 - 당나라 행정 구역	184
그림 44. 한국사 LOD 연계 개념	187
그림 45. DBpedia 서비스	189
그림 46. DBpedia에서 “서울” 개체와 다른 개체와의 연결 예시	190
그림 47. protégé Reasoner를 이용한 오류 점검의 예	191
그림 48. Gate Developer를 이용하여 관계를 표현하는 화면	193
그림 49. Virtuoso를 이용하여 RDF 트리플을 Store하는 단계	196
그림 50. Openlink Data Explore	198
그림 51. 중국 연대 대조표	203
그림 52. XSL을 적용하여 관계를 검증하는 화면	204
그림 53. XML 태그 추출기를 이용하여 XML 문서에서 SQL DB로 저장하는 화면	205
그림 54. SQL 쿼리를 이용한 RDF 트리플의 생성	206
그림 55. Pretege에서 OWL을 불러들인 화면	207
그림 56. 項羽가 劉邦을 睢水에서 대파한 사건의 RDF 그래프 표현	211
그림 57. event와 의미역을 활용한 다항 관계의 표현	212
그림 58. 실제 주어로 표현한 그래프	213
그림 59. 복수의 주어 표현	214
그림 60. 목적어가 없이 주어와 술어만 존재하는 경우	216
그림 61. 영향주에 의한 상태의 표현	217
그림 62. 朱全忠이 蔡州를 점령하니, 군세가 더욱 강성해짐을 표현하는 그래프	218
그림 63. 隋나라 文帝가 23년 동안 재위한 상태를 표현하는 그래프	219
그림 64. 匈奴의 冒頓單于가 거란을 습격한 사건을 표시하는 그래프	220
그림 65. 周市가 魏나라 公子 咎를 세워 魏王으로 삼았다 사건을 표현하는 그래프	221
그림 66. 朱全忠이 황제를 시해하고 輝王을 皇太子로 세운 사건을 표현하는 그래프	222
그림 67. 杜讓能에게 李茂貞을 토벌하는 일을 전담하도록 한 내용의 그래프 표현	223
그림 68. 동반주의 표현	224
그림 69. 사건이 발생한 장소의 표현	225

그림 70. 蘇秦이 趙나라를 떠나니 從約이 모두 와해되다의 그래프 표현	226
그림 71. 高明曜와 軍士의 목을 벤 사건의 인과 관계 표현	227
그림 72. 韓信의 event와 龍且의 event의 인과 관계의 표현	228
그림 73. 田令孜의 관작을 삭탈하여 멀리 端州로 유배보냈다는 문장의 그래프 표현	229
그림 74. 봄 2월에 朱全忠이 契丹에 使者를 보내었다의 그래프 표현	229
그림 75. 과발마를 도구로 표현	230
그림 76. 王叔文에게 사약을 하사하다를 표현한 그래프	231
그림 77. 관직을 삭탈함을 표현하는 그래프	232
그림 78. goal을 이용한 목적지의 표현	233
그림 79. 요청하는 내용을 표현한 그래프	234
그림 80. 『通鑑節要』를 대상으로 RDF 트리플 표현	235
그림 81. 『通鑑節要』 번역문을 대상으로 RDF 트리플을 표현한 예	236
그림 82. WIKIDATA에서 제공하는 문심조룡 메타데이터	238
그림 83. 五代の 인물간 관계	241
그림 84. 五代的 인물과 사건과의 관계	242
그림 85. Neo4J를 이용하여 event를 시각화한 결과	243
그림 86. 인물의 설명문을 대상으로 RDF 그래프로 표현	244
그림 87. 서명의 설명문을 RDF 그래프로 표현	245
그림 88. 馮延諤의 신분과 국적을 표현하는 RDF 그래프	249
그림 89. 朱友珪가 朱全忠을 살해한 사건을 RDF 그래프 표현	251
그림 90. 朱全忠이 蔣玄暉를 시켜 唐昭宗을 시해했음을 표시하는 RDF 그래프	254
그림 91. 馮延諤이 행위를 했는지 찾는 쿼리	257

【국문초록】

역사 자료 텍스트의 전자적 기술에 의한 지식 관계망 구현 연구

- 『通鑑節要』와 『譯註 通鑑節要』를 중심으로

그 동안 우리나라는 정보화 사업에 막대한 예산을 투입하여 『朝鮮王朝實錄』 등의 방대한 자료를 정보화하였으며, 검색 기술의 발달은 그동안 알지 못했던 새로운 지식을 쉽게 찾을 수 있게 하였다. 그러나 이러한 방대한 자료의 축적은 정보를 찾는 데 소비되는 시간 또한 점점 증가하게 하였다. 이러한 문제를 해결하기 위해서 팀 버너스 리는 ‘시맨틱 웹’을 주장하였다. 시맨틱 웹은 HTML 문서에 기반하여 데이터를 웹으로 제공하는 것이 아니라, 데이터 간의 링크를 구성하여 인간이나 기계 모두 데이터의 웹을 탐험할 수 있도록 해주는 것을 말한다.

한편 근래에는 데이터와 데이터의 관계 속에서 새로운 가치를 발견하기까지 이르게 되었다. 데이터 간의 연계와 무한 확장, 데이터 활용 주체 간의 협력, 다양한 조합·분석을 통해 가치를 창출하는 새로운 시대가 도래하게 된 것이다. 본 논문의 주제도 이러한 시대적 배경에서 역사 자료를 어떻게 하면 새로운 가치를 창출할 수 있을까 하는 논의에서 출발하였다. 그 구체적인 방법 중의 하나가 역사 자료의 연계와 통합이다. 여기서 연계는 의미적, 내용적으로 연결시켜 주는 것을 말하며, 통합은 이러한 지식을 물리적으로 결합하는 것을 의미한다.

역사 자료의 연계 및 통합은 기본적으로 LOD(Linked Open Data)을 기반으로 이루어진다. LOD로 발행한다는 것은 의미 없이 죽어있는 지식을 식별하고 ‘연결하고, 구조화시키고, 맥락화시키면서 의미 있는 것을 만들어 내는 과정’을 말한다. 기존의 LOD 발행은 대부분은 정형화되고 규칙적인 정보에 집중되어 있었다. 즉 주소 정보라든지 문화재의 메타데이터 등 전형적인 RDB 구조로 구축된 데이터만이 집중적으로 LOD로 전환이 되었으며, 역사서에 등장하는 구체적인 내용을 LOD로 표현하려는 노력은 거의 이루어지지 않았다.

본 논문은 이러한 역사서에 나타나는 본문의 내용을 LOD로 표현하려는 시도이다. 여기에 지식 관계망(Knowledge Network)은 하나의 지식 노드와 관련된 다른 노드를 서로 연결해서 분절화된 지식을 의미적으로 연결해 주는 네트워크 역할을 한다. 그럼으로써 지식의 개별 노드와 노드를 연결해서 새로운 지식으로 이동할 수 있는 소통의 교점을 마련해 준다.

지식 관계망에는 지식의 정제화와 구조화라는 두 가지 요소가 추가된다. 정제화란 정제된 지식의 寶庫인 辭典 등에서 수집한 어휘 간의 관계, 同義語, 反意語 등의 의미론적 연관 관계와 역사의 주체인 인물 간의 관계, 인물과 사건 등을 정밀하게 정제하여 RDF로 구축하는 것이다. 그리고 과거의 연구성과 등의 정제된 지식을 대상으로 하여 사람의 지혜를 이용하여 서로 간의 관계를 잘 구조화(Structureing)시킨 것을 의미한다. 이러한 지식 관계망을 통해서 기존 LOD로 해결할 수 없는 훨씬 더 다양하고 복잡한 지식으로 또한 새로운 정보의 접근도 쉬워질 것이다.

본 논문의 연구의 범위 및 방법은 다음과 같다.

첫째, 역사적 지식을 표현하기 위한 이론적, 기술적인 방법을 연구한다. 특히 이때 중요한 공백 노드(Blank node)와 중계자(Mediator), 구체화(Refication)를 이용한 표현 방법에 대해 연구하였다.

둘째, 복잡한 역사적 지식을 표현할 수 있는 온톨로지 모델에 대해 연구한다. 역사적 지식은 기존의 개체 간의 관계를 정해 놓은 모델로는 답을 수 있는 정보가 제한되어 있다. 역사서에는 사람의 이름이나 저작 등의 정적인 관계도 있지만, 본문의 대부분은 동사나 형용사로 표현해야 하는 동적인 관계로 구성되어 있다. 그런데 이러한 관계는 미리 온톨로지로 설계해 놓지 않으면 답을 수 없다. 그렇다고 해서 국어사전에 등록된 모든 동사나 형용사를 관계로 등록할 수는 없다. 등록한다고 해도 하나의 행위에 관련 있는 부가적인 정보를 표현할 수도 없다. 예를 들어 행위가 발생한 시간이나 장소 등을 동시에 표현하기는 일반적인 모델로는 할 수 없다는 말이다. 이 문제를 해결하기 위해 본 논문에서는 統辭論에서 사용하는 意味役 개념을 도입하는 방안을 연구하였다. 그 결과 기존의 모델로 불가능했던 문장을 RDF 트리플로 표현하는 것이 큰 어려움 없이 표현할 수 있게 될 것이다.

셋째, 역사적 지식을 연계 및 통합하는 구체적인 방법으로 LOD를 제안하고 그 온톨로지 모델을 제시하였다. 역사자료에 적합한 온톨로지 모델을 설계하기 위해서 『通鑑節要』를 대상으로 전체 데이터에 중요한 개체와 관계에 마크업을 하고 이를 RDF로 변환하는 과정을 반복하였다. RDF로 구축하기 위해서는 개체 간 통합 작업이 필요한데 이를 위해 개체 간에 동일한 대상일 경우에 대표명칭으로 통합하였다. 사건은 역사적 지식을 기술하는 데 매우 중요한 역할을 한다. 사건은 실질적으로 ‘공격하다, 죽이다, 점령하다, 대과하다’ 등의 인간의 행위 혹은 상태로 구성되어 있다. 이러한 행위와 상태를 클래스로 표현한다. 그리고 하나의 사건은 관련된 가해자, 피해자 등의 인물, 행위, 발생 장소, 원인과 결과, 사건이 발생한 시간과 종료된 시간, 사건의 선후 관계, 후행 사건 등을 표현하는 방법을 연구하였다.

넷째, 역사적 지식을 더 깊이 있고, 유기적으로 연계하기 위한 방법으로 지식 관계망을 제안한다. 역사적 지식을 LOD를 구축할 수 있는 모델링을 통해서 연계 및 통합의 방법을

제시한 기존의 연구에서 더 나아가, 이러한 지식을 좀 더 심도 있게 확장할 필요가 있다. 이를 위해서 인간의 연구 결과로 축적된 정제된 지식이 필요하며, 본 연구에서 제시하는 지식 관계망이 그 역할을 해줄 것이다.

지식 관계망에 있어 개체와 개체 사이에 관계는 크게 혈연 관계처럼 잘 변하지 않는 정적인 관계와 官歷처럼 자주 변동되는 동적인 관계로 구분할 수 있다. 인물 간의 관계는 크게 혈연 관계, 혼인 관계, 사회적 관계로 구분하여 세부 프로퍼티를 구성하였으며, 역사적 지식을 세밀하게 표현할 수 있도록 행위 동사를 클래스로 도출하였다.

어휘 간 관계는 본문 내용의 표현과는 관계없이 순수하게 의미적인 연관 관계를 표현해야 하는 경우가 있다. 의미적 연관 관계는 同義語, 類義語나 反意語, 계층 관계, 내용적 연관 관계 등을 의미하는 것으로써 역사적 지식을 표현하는 데도 적용할 수 있지만 지식 관계망을 구성하는 데에도 중요한 역할을 한다.

그리고 역사 자료의 통합과 연계에서 서력과 연호, 묘호, 시호로 표현하는 연도의 매핑 또한 중요하기에, 이를 표현할 수 있는 프로퍼티를 구성하였다.

본 논문의 의의는 다음과 같다. 역사서에서 본문의 내용을 대상으로 어떤 행위를 기본 단위로 하여 LOD를 발행하기 위해서는 역사서에 어떤 내용이 담겨 있으며 어떤 구성을 가지고 있는지 분석해야 한다. 그 분석에 있어 내용을 이루는 문장이 간단하여 사람이 쉽게 이해할 수 있다고 하여 기계도 자연스럽게 이해하는 것은 아니다. 아주 단순한 내용 하나를 기술하기 위해서도 많은 구문이 필요하기도 하다. 본 논문은 이러한 점에서 역사서의 내용 자체를 구체적으로 표현하는 방법을 연구하고 분석했다는 점에서 의미가 있다고 하겠다.

I. 序論

1. 연구의 배경 및 목적

그동안 우리나라는 정보화 사업에 막대한 예산을 투입하여 『朝鮮王朝實錄』, 『承政院日記』, 동양 고전번역서 등 방대한 자료를 정보화하는 등 외연적으로는 눈부신 성과를 내었다. 또한 검색 기술의 발달로 그동안 알지 못했던 새로운 지식을 쉽게 찾을 수 있는 획기적인 계기를 마련해 주었다. 그러나 자료가 축적될수록 역설적으로 자료를 찾는 데 걸리는 시간은 점점 증가하게 되었다. 이 때문에 기관별로 축적된 방대한 자료를 한 곳에서 찾아보고자 하는 요구 또한 증가하게 되었다.

그 결과로 등장한 방법이 메타데이터에 의한 통합 검색이다. 역사 분야에서는 국사편찬위원회를 중심으로 한국역사정보통합시스템¹⁾을 구축해 30개 기관에서 제공한 메타데이터를 통합 검색할 수 있는 서비스를 제공하고 있다.²⁾ 그러나 메타데이터가 가지는 한계 등으로 원하는 자료에 접근하기는 쉽지 않고, 정확성도 떨어져 점차 활용의 한계가 드러나기 시작하였다. 정보의 축적과 기술의 발달로 인해 직접 도서관을 방문하여 자료를 찾는 불편함은 해결되었지만, 이용자는 여전히 관련된 지식을 찾기 위해서 끝없이 정보의 바다에서 유랑해야만 한다.

미국 NSF(미국 국가 과학 재단)의 조사에 의하면 미국 연구자들은 전체 연구 개발 활동의 절반 이상(50.9%)을 관련 정보를 수집하는 데 소요하고 있으며, 일본 과학기술청 역시 일본의 연구자와 기술자들이 하루 업무 중 20% 이상을 정보 수집에 할애하고 있다고 밝힌 바 있다.³⁾ 그렇기 때문에 이제는 무엇보다도 원하는 정보를 정확하게 찾아주는 것이 중요하다. 단순히 찾고자 하는 정보를 보여주는 것 뿐만 아니라 그와 관련된 정제된 지식을 획득할 수 있어야 한다.

이러한 문제를 해결하기 위해서 팀 버너스 리(Tim Berners-Lee)는 ‘시맨틱 웹(semantic web)’을 주장하였다. 시맨틱 웹은 HTML 문서에 기반하여 데이터를 웹으로 제공하는 것이 아니라, 데이터 간에 링크를 만듦으로써, 인간이나 기계 모두 데이터의 웹을 탐험할 수 있도록 해주는 것을 말한다. 이러한 시맨틱 웹을 구현하는 핵심 기술 중의 하나가 바로 LOD(Linked Open Data)이다. LOD는 기계 가독성이 있는 구조화된 데이터를 생성해서 오픈된 데이터로 서로 연결해(Link) 가치를 만들어 나가는 것을 말한다.⁴⁾ 시맨틱 웹의 링크드

1) 한국역사정보통합시스템, 국사편찬위원회, <http://www.koreanhistory.or.kr>

2) 한국역사정보통합시스템 사이트에 의하면 2017년 7월 8일 현재 30개의 기관이 참여하고 있다.

3) 「국가과학기술 지식정보 유통서비스를 위한 시맨틱 기술」, 『지식정보인프라』 26호 2007, 16쪽.

데이터를 통해 유용한 데이터를 얻게 되면, 그 데이터에 관계된 데이터로 계속해서 항해가 가능하다.

지금까지의 지식은 독립적이고 완결된 지식으로써 존재했다. 과거에는 지식이 지식인과 합체였고 그것의 사용자는 생산자와 만나거나 결합되기 힘들었다. 그러나 최근의 지식은 응용하기 쉬운 최소 단위로 분리되어 모듈화된다. 링크와 하이퍼텍스트라는 미디어 형식은 정보와 지식의 상호 연결성과 연관성을 확대하였다.⁵⁾ 이러한 과정에서 어느 순간부터 데이터 간 관계 속에서 기존 지식에서 찾지 못했던 새로운 지식이 발견되고 이러한 지식에서도 가치가 발생한다는 것을 깨닫게 되었다. 데이터를 공유하고 의미적으로 연계시킴으로써 이전에 보지 못했던 새로운 영역이 열리게 된 것이다. 그 결과 데이터 간의 연계와 무한 확장, 데이터 활용 주체 간의 협력, 다양한 조합·분석을 통해 가치를 창출하는 새로운 형태의 데이터 경제 시대를 맞이하게 되었다.

본 논문의 주제도 이러한 시대적 배경에서, 어떻게 하면 역사 자료로 새로운 가치를 창출할 수 있을까에서 출발하였다. 그 구체적인 방법 중의 하나가 역사 자료의 연계와 통합일 것이다. 여기서 연계는 의미적, 내용적으로 연결시켜 주는 것을 말하며, 통합은 이러한 지식을 물리적으로 결합하는 것을 의미한다.

역사 자료의 연계 및 통합은 기본적으로 LOD를 기반으로 이루어진다. LOD는 링크드 데이터와 오픈 데이터의 결합을 의미하는 것으로, 서로에 대해 관계가 있음을 알리는 데이터의 집합을 링크드 데이터라고 하며, 그 링크드 데이터가 인터넷 상에서 공개되어, 누구라도 그 연결을 확장해 나갈 수 있는 것을 링크드 데이터라고 한다.⁶⁾ LOD로 발행한다는 것은 의미 없이 죽어있는 지식을 식별하고 ‘연결하고, 구조화시키고, 맥락화시키면서 의미 있는 것을 만들어 내는 과정을 말한다. 역사 자료는 특히 다른 분야보다 어느 하나의 지식이 독립적으로 존재하기보다는 서로 유기적인 관계를 맺고 전개가 된다. 하나의 사건에는 원인과 결과가 있으며, 주체가 있고 객체가 있기 마련이다. 역사적 지식을 LOD로 잘 발행해 놓으면 사건의 인과 관계나 선후 관계를 명확하게 살펴볼 수 있으므로 지식을 확장하여 새로운 지식을 생산해 낼 수 있을 것이다.⁷⁾

이를테면 『조선왕조실록』을 대상으로 본문 중에 나타나는 정보에 하나하나 의미를 부여한다면 이 정보와 다른 자료를 연결하여 많은 새로운 지식을 만들어 낼 수 있다. 예를들어

4) 한국정보화진흥원, 『LOD 기반의 데이터 관리 패러다임 전환 전략』, 『IT & Future Strategy 보고서』, 1, 2014, 6쪽.

5) 백옥인, 『디지털 데이터·정보·지식』, 커뮤니케이션북스, 2013, 23쪽.

6) 김현·임영상·김바로 공저, 『디지털 인문학 입문』, 한국외국어대학교 지식출판원, 2016, 148쪽.

7) LOD는 사람을 대상으로 한다가 보다 기계를 대상으로 한다고 할 수 있다. 그러나 어떤 식으로든 최종 소비자는 사람이기 때문에 사람이 원하는 자료를 만들어 내야 하는 것이다.

역사 자료에 관직에 임명되거나 면직된 정보를 정리하였다면 관력 정보가 자동으로 만들어지는 것이다.

그러나 이렇게 역사적 지식을 LOD로 발행했다 하더라도 지식의 연계 부분에서는 아직 제한적일 수밖에 없다. 역사적 지식을 RDF로 표현했다 뿐이지, 지식 자체를 구조화하여 외부에서도 쉽게 연계할 수 있는 구조가 아직은 아니다.

그 이유로는 첫째, LOD는 아직 제한적인 범위에서만 사용되고 있기 때문이다. 국내에서 국가 주도로 LOD로 정보화하는 작업을 계속해서 진행하고 있지만, 예산 등의 이유로 LOD가 폭넓게 발행되고 있지 못하다. 특히 역사 자료는 더더욱 그렇다. 비록 한국사 LOD가 발행되었으나 시범 사업으로 추진된 것이고 그 이후에는 별다른 성과를 내고 있지 못하고 있다. 둘째, LOD 구축이 대부분은 정형화되고 규칙적인 정보에 집중되어 있기 때문이다. 즉 주소 정보라든지 문화재의 메타데이터 등 전형적인 RDB 구조로 구축된 데이터가 집중적으로 LOD로 전환이 되었으며⁸⁾, 역사서에 등장하는 구체적인 내용을 LOD로 표현하려는 노력은 거의 이루어지지 않았다.

LOD를 통해서 『조선왕조실록』이 언제, 누구에 의해서 간행되었는지, 책의 크기는 어느 정도인지에 대한 단편화된 지식은 얻을 수 있으나, 『조선왕조실록』에서 어떤 사건이 있었는지, 그 사건과 관련된 사람은 누구인지, 구체적으로 어떻게 사건이 진행되었는지 등에 대한 지식을 얻기는 힘들다.⁹⁾ 즉, LOD를 통해서 주변적인 정보는 확인할 수 있으나 내용 그 자체에 접근하기는 쉽지 않다는 말이다. 그러나 이용자는 역사서에 등장하는 사건을 중심으로 왜 그런 사건이 발생하였으며, 원인이 무엇이고 그 결과는 어떻게 되었으며, 역사적으로는 어떻게 평가되는지 등의 종합적이고 완결된 지식을 원한다.

역사적 지식을 LOD로 표현하기 위해서는 좋은 온톨로지 모델이 필요하다. 역사적 지식을 일정한 원칙에 의해서 구축하지 않으면 지식으로서 활용성이 떨어지기 때문이다. 본 논문의 목적이 바로 여기에 있다. 즉 역사적 지식을 잘 담을 수 있는 온톨로지 모델을 제시하고자 한다. 이 온톨로지 모델은 LOD의 발행뿐만 아니라 지식 관계망 구조를 수용할 수 있도록 설계하였다.

지식 관계망(Knowledge Network)은 하나의 지식 노드와 관련된 다른 노드를 서로 연결해서 분절화된 지식을 의미적으로 연결해 주는 네트워크 역할을 한다. 지식 관계망은 지식의 개별 노드와 노드를 연결해서 새로운 지식으로 이동할 수 있는 소통의 교점을 마련해 준

8) 유로피아나(Europeana)는 원천 자료의 소장 기관으로부터 디지털화된 콘텐츠의 메타데이터만을 제공받아 이들을 의미적 연관을 가지고 서비스될 수 있도록 하였다. 방대한 문화재에 대한 관련 정보를 서로 연계시켜 지식의 확장에는 많은 도움을 주고 있다.

9) 한국사 LOD에서 사건의 일부를 보여주고 있다.

다.¹⁰⁾ 역사적 지식을 LOD로 발행하면 상당수의 지식을 서로 연결할 수 있겠지만 모든 역사적 지식을 LOD로 발행할 수 없다는 현실적인 문제로 인해 지식 관계망이 필요한 것이다. 지식 관계망에는 정제화와 구조화라는 두 가지 요소가 추가된다. 정제화란 정제된 지식의 寶庫인 辭典 등에서 수집한 어휘 간의 관계, 同義語, 反意語 등의 의미론적 연관 관계와 역사의 주체인 인물 간의 관계, 인물과 사건 등을 정밀하게 정제하여 RDF로 구축하는 것이다. 즉 정제된 정보를 이용하여 사람의 지혜를 이용하여 서로 간의 관계를 잘 구조화(Structuring)시킨 것을 의미한다. 향후 지식 관계망은 내부의 지식과 외부의 지식을 소통시키는 역할을 한다. 지식 관계망을 통해서 기존 LOD로 해결할 수 없는 훨씬 더 다양하고 복잡한 지식으로 접근이 가능하게 될 것이며 새로운 정보로 접근할 수 있게 될 것이다.

과거에는 도서관 등에서 접근이 물리적으로 차단된 형태를 사일로(silo)라고 하였으나 이제는 의미적으로 다른 자료와 연계되지 않는 지식을 사일로라고 불러야 할 것이다. 이러한 의미적 사일로를 해결하는 방법이 바로 지식 관계망이 되는 것이다.

2. 연구의 범위 및 방법

역사적 지식을 LOD로 발행하여 서로 연결한다는 것은 지식을 RDF 트리플로 만든다는 것을 의미한다. RDF 트리플은 지식을 사람뿐만 아니라 컴퓨터가 쉽게 처리할 수 있도록 웹에 있는 자원을 주어(Subject)-술어(Predicate)-목적어(Object)로 표현한 논리적인 모델이다. 이러한 모델은 도서관의 메타데이터나 서지 정보 등을 표현하는데는 유용하나 일반적인 문장을 표현하는데는 많은 제약이 있다. 예컨대 ‘項羽가 義帝를 죽이다’와 같은 간단한 문장은 RDF 트리플로 표현할 수 있다. 그러나 여기에서 ‘죽이다’는 행위가 발생한 사건에 시간과 공간 같은 가장 기초적인 정보를 추가하는 것조차 쉽지 않다. RDF 트리플에서 허용하지 않는 구조이기 때문이다. 즉 행위 자체가 발생한 시간과 공간은 단순한 二項 관계로는 표현이 불가능하다.

이를 표현하기 위해서 공백 노드(Blank node)를 사용해서 표현하는 방법과 중계자를 이용해서 표현하는 방법 등이 연구되었다. 공백 노드는 이름을 갖지 않는 노드를 말하며 내부적으로 노드와 노드 사이에 공백 노드를 배치하여 정보를 좀 더 확장적으로 표현하는 방법이다. 아래는 ‘項羽’와 ‘義帝’를 직접 연결하지 않고 중간에 공백 노드를 두고 ‘項羽’와 ‘죽이다’라는 행위를 연결하고 다시 공백 노드를 중심으로 장소와 대상 발생시기를 연결하는 것이다. 이들 관계는 개별적으로는 모두 주어-술어-목적어 관계를 유지하고 있다.

10) 김현·윤종웅·안성수, 「동양고전어휘정보화의 방법 및 활용 방안」, 2013, 11~12쪽.

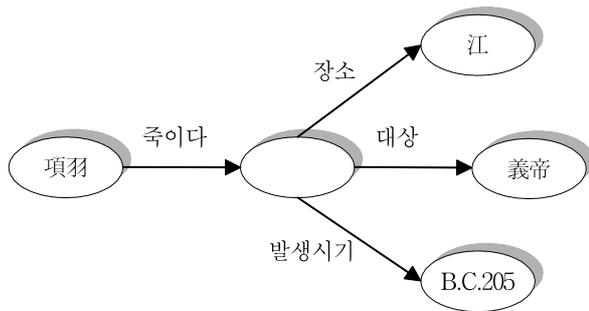


그림 1. 공백 노드를 이용한 시간과 공간의 표현

그러나 이런 방법은 약간의 부가 정보를 표현할 수는 있으나 역사서에 나오는 다양한 행위를 공백 노드를 이용하여 표현하기에는 아직도 많은 제한이 있다. 지금까지 대부분의 온톨로지 설계는 사전에 정해진 관계만을 수용할 수 있었다. 만약 미리 정해 놓은 관계가 아닌 새로운 형태의 관계가 나타나면 매번 관계를 추가해야 했다. 실제로 CIDOC 표준은 버전이 바뀔 때마다 새로운 관계가 추가되는 것을 확인할 수 있다. 만약 이런 방식을 역사서의 본문을 RDF로 표현한다면 어쩌면 국어사전에 등재된 모든 어휘를 프로퍼티로 등록해야 할지도 모른다. 역사서의 특성상 전문어가 많다는 점을 생각하면 국어사전보다 더 많은 관계를 설정해야 할 것이다.

또한 현재까지 공백 노드를 활용하여 문장을 일관성 있게 표현할 수 있는 연구가 되어 있지 않다. 역사서에 담긴 복잡다단한 내용에서 어떤 문장을 표현하든지, 어느 누가 표현하든지 일관성 있게 적용시킬 수 있는 표준이 필요하다. 이러한 표준이 마련되지 않으면 역사서의 본문 내용을 LOD로 통합하는 건 애초부터 불가능하기 때문이다.

역사서에는 사람의 이름이나 저작 등의 정적인 관계도 있지만, 본문의 대부분은 동사나 형용사로 표현해야 하는 동적인 관계로 구성되어 있다. 그런데 이러한 관계는 미리 온톨로지로 설계해 놓지 않으면 답을 수 없다. 그렇다고 해서 국어사전에 등록된 모든 동사나 형용사를 관계로 등록할 수는 없다. 등록한다고 해도 하나의 행위에 관련 있는 부가적인 정보를 표현할 수도 없다. 예를 들어 행위가 발생한 시간이나 장소 등을 동시에 표현하기는 일반적인 모델로는 할 수 없다는 말이다. 본 연구에서는 이러한 문제점을 해결하기 위해 統辭論에서 사용하는 意味役 개념을 도입하는 방안을 연구하였다. 그 결과 기존의 모델로 불가능했던 문장을 RDF 트리플로 표현하는 것이 큰 어려움 없이 표현할 수 있게 될 것이다.

다음으로 지식 관계망을 이용한 역사 자료의 연계 및 통합을 위해서 본 논문에서는 크게

네 부분에 대한 연구를 수행한다.

첫째, 역사적 지식을 표현하기 위한 이론적, 기술적인 방법을 연구한다. 역사적 지식을 표현하기 위해서는 RDB에 저장된 관계형 데이터가 아닌 자연어 처리를 염두에 두고 설계를 해야 한다. 이러한 설계에는 일반적인 주어-서술어-목적어 관계의 이항 논리로는 해결할 수 없는 것이 많다. 그렇기 때문에 적절한 중개자(mediator)¹¹⁾, 공백 노드(Blank node)와 구체화(Refication) 과정을 거쳐야 한다. 본 연구에서는 이러한 공백 노드와 구체화 방안을 『通鑑節要』에 나타나는 역사적 지식을 대상으로 RDF 그래프로 표현하는 방법을 연구하였다. 그런데 RDF는 OWL로 변환할 수는 있지만, 문법 자체가 완전히 같은 것은 아니다. 예를 들어 RDF가 'rdf:Description'에 의해서 노드를 표현하는 데 비해 RDFS부터는 클래스(Class)라는 개념을 도입하였으며, OWL에서는 직접 'Class'명으로 표현하게 되었다.¹²⁾ 역사적 지식을 OWL로 기술하는 데 필요한 기본적인 사항을 OWL 표준을 참조하여 이론적으로, 기술적으로 연구한다.

둘째, 역사적 지식을 연계 및 통합하는 구체적인 방법으로 LOD를 제안하고 그 구체적인 온톨로지 모델을 제시한다. LOD로 지식을 구현하기 위해서는 온톨로지 모델이 필요하다. 역사서를 통합하기 위한 온톨로지 모델링을 위해서는 먼저 역사서에서 의미 있는 클래스와 프로퍼티를 도출해야 한다. 클래스와 프로퍼티, 그리고 단순한 리터럴(문자열)¹³⁾로 표현해야 할지를 결정해야 한다. 이러한 판단의 기준을 마련하기 위해 온톨로지 구축 방법론을 참조하였다. 클래스를 도출하기 위해 본 논문에서는 실증적인 방법을 적용하였다. 모든 역사 자료 전체를 대상으로 분석할 수 없기 때문에 본 연구에서는 중국사를 다룬 역사서 중 조선 시대에 가장 널리 읽혔던 『通鑑節要』를 대상으로 본문 중에 나타나는 개체명을 직접 마크업하는 방식으로 개체명을 도출하였다. 『通鑑節要』를 선택한 이유는 『通鑑節要』는 번역문뿐

11) 중개자(mediator)는 다항 관계에서 공백 노드를 사용하여 주어와 목적어 관계를 연결해 주는 방식에서 트리플을 적절하게 표현할 수 있는 클래스 이름을 부여하여 그 노드와 목적어를 연결해 주는 역할을 한다.

12) 이러한 내용은 온톨로지 개론서에는 대부분 나와 있지 않다. 그러나 실제로 사용하는 도구에서는 OWL 문법을 적용하기 때문에 굉장히 혼란스럽다.

13) 리터럴(Literal)은 문자열이라고도 하며 '문자로 작성된 것'을 말한다. 즉 URL 참조가 아닌 실제 데이터인 문자열을 가리킨다. 본 논문에서는 앞으로 모든 용어를 '리터럴'이라는 용어로 표현한다. 이 리터럴에는 의미를 명확하게 표현하기 위해 데이터 형을 부여할 수 있다. RDF/XML에서는 rdf:datatype 속성에 데이터 형의 URI를 지정할 수 있다. RDF1.1 Concepts and Abstract Syntax에 의하면 datatype은 XML Schema에서 제공하는 datatype을 사용한다.(<https://www.w3.org/TR/rdf11-concepts>). W3C에서 제공하는 XML Schema Part 2: Datatypes Second Edition에 의하면 XML Scheme에 내장된 datatype을 <https://www.w3.org/TR/xmlschema-2>에서 확인할 수 있다.

만 아니라 원문을 수록하고 있기 때문에 번역문에 사용된 서술어를 바탕으로 원문에 사용된 한문의 서술어를 연계시키는데 용이하기 때문이다. 개체명을 도출하기 위해 『通鑑節要』의 原文과 翻譯文¹⁴⁾에 개체명이 될 수 있는 요소를 모두 마크업하였다. 그리고 다른 논문이나 표준, 연구 등을 참조로 하여 개체명으로 의미가 있는 것을 도출했다. 개체명은 일종의 문맥 요소로서 그 자체로도 의미가 있지만 외부 데이터로 연계를 위한 접근점이 될 수 있다는 점에서 매우 중요하다.

클래스와 클래스, 클래스와 리터럴을 연결하는 것을 프로퍼티(Property, 속성)¹⁵⁾라 한다. 이러한 프로퍼티를 설계하는 것은 관계를 표현함으로써 지식이 완결되기 때문에 역사적 지식을 표현하기 위해서는 프로퍼티를 잘 설계해야 한다. 역사적 지식은 단순히 정적인 사건 뿐만 아니라 동적인 행위로도 이루어져 있다. 본 논문에서는 이러한 점을 주목해 역사적 지식 중 행위를 프로퍼티로 설계하는 방법을 연구하였다. 역사적 지식은 결정된 지식과 변동하는 지식으로 구분할 수 있다. 결정된 지식이라는 것은 인물의 字나 號 등과 같이 쉽게 바뀌지 않는 지식을 말하며, 변동하는 지식이라는 것은 관력이나 행위 등 역사서에서 시간에 따라 어떤 내용이 변동이 발생하는 지식을 말한다.¹⁶⁾ 바로 이러한 동적인 지식을 표현하기 위해서 행위를 표현할 수 있는 프로퍼티가 필요하다. 하나의 역사적 사건은 사건 하나로써 의미를 파악할 수 있는 것이 아니라 사건을 둘러싼 사건의 가해자, 피해자, 구체적인 행위, 원인과 결과, 발생한 시간, 장소, 사건의 선후 관계 등을 종합적인 관계 속에서 이해해야 한다. 본 연구에서는 바로 이러한 사건의 종합적인 기술을 위한 온톨로지 모델을 연구하고 설계하였다.

그리고 그 구체적인 해결 방법으로 본 논문에서는 모든 행위를 이벤트(event)로 보고 이들 행위와 다른 개체와의 관계를 통사론에서 사용하는 意味役을 도입하여 표현하고자 하였다. 여기서 意味役은 한 문장이 완성되기 위해 꼭 필요한 문장 성분과 그 문장 안에 있는 명사나 형용사 등의 슬어가 맺고 있는 의미적 관계를 말한다.¹⁷⁾ 이러한 意味役으로 관계를 기술할 경우 역사서에 나타나는 문장을 아무런 제한 없이 표현할 수 있을 것으로 기대된다.

14) 成百曉 역, 『譯註 通鑑節要』1~9, 전통문화연구회, 2006. 전통문화연구회에서 『通鑑節要』를 9책으로 완역을 하였으며, 이 자료는 현재 온라인으로 서비스되고 있다. 『譯註 通鑑節要』에는 원문과 번역문이 모두 실려 있으므로 이 자료를 이용하여 RDF 트리플을 구축하였다.(<http://db.cyberseodang.or.kr/front/main/main.do>)

15) Property는 속성, 슬롯이라는 명칭이 사용되기도 한다.

16) 여기서 변동이 된다는 것은 역사적 사실 자체가 바뀐다는 의미가 아니라 시간에 따라 행위나 내용이 바뀐다는 의미이다. 사람을 따지면 관직에 나가서 점차 승진하면서 관직이 계속해서 바뀌게 된다. 관직을 역임한 사실은 변함이 없으나 관직이라는 관점에서 보면 계속해서 바뀌는 것이다.

17) III-4-3) 의미역에서 의미역의 종류에 대해서 상세하게 다루고 있다.

이들 意味役을 통한 행위의 표현은 『通鑑節要』 전체 50권 중에서 48권, 49권, 50권 등 모두 3책에 대해서 적용하였다. 이 기간은 시대적으로 보면 唐 宣宗이 즉위한 846년부터 宋 태조가 즉위하기 직전인 960년까지 1백 년이 넘는 기간 동안이다.

셋째, 역사적 지식을 더 깊이 있고, 유기적으로 연계하기 위한 방법으로 지식 관계망을 제안한다. 기존의 연구가 역사적 지식을 LOD를 구축할 수 있는 모델링을 통해서 연계 및 통합의 방법을 제시했다면, 이러한 지식을 좀 더 심도 있게 확장하기 위해서는 인간의 연구 결과로 축적된 정제된 지식이 필요하다. 지식 관계망은 이러한 인간의 연구 성과를 도입하여 어휘 간의 관계를 의미적으로 내용적으로 엮어서 지식을 확장시키는 지식 네트워크이다. 정제된 지식은 주로 辭典을 의미하며, 지식을 제공하고 있는 사이트와의 연계를 포함한다.

이러한 세 가지 연구를 하는 데 있어서 연구의 대상 자료를 『通鑑節要』를 선택한 이유는 다음과 같다. 첫째, 『通鑑節要』는 조선 시대에 사대부들이 한문 학습 과정에서 거의 모든 사람이 읽어온 주요 교재 역할을 하였다는 점을 들 수 있다. 각종 문집이나 『朝鮮王朝實錄』 등에서 『通鑑節要』에서 나온 故事成語 등이 수없이 등장하고 있다. 『通鑑節要』가 사료로서의 좋고 나쁨을 떠나서 현실적으로 조선 시대 사대부들의 역사 인식을 알 수 있는 중요한 자료라는 점을 고려하였다. 『通鑑節要』는 조선 시대 사대부들과 동일한 관점에서 중국사를 이해하고 그 바탕에서 지식의 확장이 필요하기 때문이다. 둘째, 중국사를 통사적으로 이해하는데 『通鑑節要』가 유용하기 때문이다. 『通鑑節要』는 周나라 威烈王 23년(BCE 403)부터 後周 顯德 7년(960년)까지 1363년 동안의 긴 기간 동안의 중국 역사를 다루고 있다. 이는 춘추 시대를 제외하고 전국 시대부터 宋나라가 통일하기 직전까지의 시기이다. 또한 『通鑑節要』가 『資治通鑑』을 節錄하였기에 내용을 압축적으로 담고 있기 때문이다. 셋째, 『通鑑節要』는 편년체 史書로써 왕조, 연도순으로 사건을 기술하였기 때문에 시간 흐름에 따른 사건의 추이를 살펴보기 좋은 자료이기 때문이다. 특히 이 사서는 사건에 대한 인과 관계가 비교적 충실하게 제시되어 있기 때문에 대중적인 흥미를 유발할 수 있는 자료이다.

넷째, 동아시아 한자문화권 국가에서 서로 간에 지식의 소통이 원활하게 이루어질 수 있도록 하는 가능성을 모색하였다. 한국, 일본, 중국 등의 국가들은 근대 이전에 한문을 이용해 역사를 기록했다는 특징이 있다. 그러면서 이들 자료를 학습하거나 활용할 경우에는 번역문을 이용하였다. 그렇기 때문에 역사 사료에서 한문 원전에 쓰여진 표현과 그 의미를 적절하게 현대적으로 해석했을 때 쓰는 표현 간의 상관 관계에 대해서도 우리가 지속적으로 연구할 필요가 있다. 구체적으로 event 노드를 비롯해 한문 원전에 사용된 어휘를 번역문과 연결시키는 작업을 진행하였다. 한문 원문에 대한 번역은 각국의 언어별, 시대별로 새롭게 만들어 지고 있으며, 번역하는 사람에 따라 그 표현이 달라지기 마련이다. 이런 번역에 사용된 상이한 표현들을 지속적으로 LOD로 축적하면 향후 자동 번역에도 큰 도움이 될 것

이다. 이때 중요한 것은 어휘의 URI 부여이다. 본 연구에서는 원문에 사용된 한자짜리 한자는 『漢韓大辭典』의 의미 구분을 따르도록 하였다. 그렇지만 제한된 시간이라는 현실적인 한계로 인해 우선 『通鑑節要』와 『譯註 通鑑節要』를 대상으로 하였다.

온톨로지를 설계하기 위한 구체적인 방법론은 EOE(Evolving Ontology Engineering)를 응용하였다. EOE 방법론은 진화형 프로토타입(evolutionary prototype)을 바탕으로 하고 있다. 진화형 프로토타입은 시간 경과에 따라 요구사항이 분명해지면서 온톨로지를 점진적으로 구축할 수 있는 견고한 아키텍처를 제공한다. 진화형 프로토타입은 일회용 프로토타입보다 더 많은 시간이 소요되지만, 신뢰성 있고 쉽게 확장할 수 있기 때문에 온톨로지를 구축하는데 적합하다.¹⁸⁾ 쉽게 말하면 본문 중에 어떤 내용이 있는지 직접 확인하면서 개체명을 도출하고, 도출된 개체명을 프로퍼티에 의해 서로 연결하면서 맞지 않는 것은 수정하는 작업을 반복하면서 온톨로지 모델을 도출하는 방식이다. 온톨로지 설계는 OWL을 기본 문법으로 하였으며 활용 도구는 스탠퍼드대학교에 무료로 제공하는 Protégé^{5.219)}를 사용하였다.

3. 논문의 구성

본 논문의 구성은 다음과 같다. 본 논문에서 제안한 역사 자료를 연계 및 통합하기 위한 온톨로지 모델은 역사적 지식을 행위와 상태를 프로퍼티가 아닌 클래스로 표현하는 방법을 연구하고 있다. 이러한 연구는 기존에 다른 연구에서 찾기 어려운 것으로 이를 구현하기 위해 필요한 제반 기술과 연구 현황을 II장에서 다루고 있다.

III장에서는 역사 자료를 연계 및 통합 방안으로 온톨로지 모델을 제시한다. 역사 자료에 적합한 온톨로지 모델링을 하기 위해서는 먼저 클래스와 프로퍼티를 도출해야 한다. 클래스를 도출하기 위해서는 역사서에 어떤 개체가 담겨 있는지 확인해야 하며 어떤 개체가 유의미한지를 확인해야 한다. 이를 위해서 본 논문에서는 톱다운(Top-down) 방식과 보텀업(bottom-up) 방식을 함께 적용하였다. 우선 클래스의 대상이 되는 개체를 정하기 위해 먼저 기존 연구서와 지침, 선행 연구를 참고로 하여 개체명을 도출하고, 도출된 개체명의 유효성을 확인하기 위해 전통문화연구회에서 온라인으로 서비스하고 있는 『譯註 通鑑節要』의 원문과 번역문에 마크업하였다. 이를 기반으로 온톨로지를 설계하였다. 클래스와 프로퍼티를 설계할 때 지식의 통합을 위해 행위를 나타내는 동사와 상태를 나타내는 형용사에 주목하였다. 이들 행위와 상태를 기반으로 하는 온톨로지는 기존 연구에서 담을 수 없었던 다양한 역사적 지식을 담을 수 있을 것이다. 그리고 역사의 주체인 인물의 대표명칭 구현 방안과

18) 온톨로지연구소, 「국가지식정보 온톨로지 표준개발」, 2006, 94쪽.

19) Protégé는 스탠포드 대학교에서 개발한 온톨로지 편집기로 2017년 4월 현재 Protégé5.2.0까지 나와 있다.(Protégé, 스탠포드 대학교, <http://Protégé.stanford.edu/products.php#desktop-Protégé>)

동일한 구간을 나타내는 시간 간의 관계를 엮어주는 방법을 중점적으로 연구하였다. 또한 프로퍼티에서는 하나의 긴 구간을 가지는 사건을 구성하는 세부적인 사건 혹은 행위를 표현하는 행위 프로퍼티의 종류와 구체적인 RDF로 표현하는 방법을 집중적으로 연구하였다.

IV장에서는 마지막으로 LOD 구축을 통해 해결할 수 없는 지식의 확장을 위해서 정제된 지식을 보완할 수 있는 지식 관계망 개념과 구조를 제안하였으며, LOD 발행 절차를 태깅 단계로부터 최종 발행 과정까지 구체적인 방안을 연구하였다. V장에서는 연구의 한계와 제한점 등을 제시하였다.

II. 이론적 배경 및 선행 연구

본 장에서는 역사적 지식을 표현하기 위해 본 연구와 관련된 온톨로지의 이론을 알아보고, 역사적 지식을 LOD로 발행하는데 참고가 되는 주요 사이트에 대해서 분석한다. 아울러 역사적 지식의 핵심이 되는 사건 온톨로지의 연구 현황을 살펴 본다.

1. 이론적 배경

가. 시맨틱 웹(Semantic Web)과 온톨로지(Ontology)

시맨틱 웹(Semantic Web)이란, World-wide-web을 창시한 팀 버너스 리가 창안한 차세대 웹 기술로, 현재의 웹처럼 사람이 눈으로 보고 이해하는 웹이 아닌 기계(컴퓨터)가 이해할 수 있는 웹을 의미한다. 팀 버너스 리는 시맨틱 웹을 기존의 웹과 별개가 아니라 현재의 웹을 확장한 것으로 정의하였다.²⁰⁾ 컴퓨터가 웹을 이해하기 위해서는 데이터와 데이터를 식별하고, 이들 데이터를 서로 명시적으로 연결해 주어야 한다. 같은 언어를 쓰는 사람끼리 의사소통이 가능하듯이, 의미가 통하는 웹이 만들어지기 위해서는 일정한 데이터 기술문의 어휘에 관한 약속을 만들고 이를 지켜야 한다. 이와 같은 공통된 약속을 온톨로지라는 이름으로 부른다.²¹⁾

시맨틱 웹에 대한 기술 연구는 크게 웹의 표준을 담당하고 있는 W3C를 중심으로 한 OWL(Web Ontology Language) 기반의 온톨로지(Ontology) 기술과 ISO(International Organization for Standardization)²²⁾를 중심으로 한 Topic Maps 기반의 온톨로지 기술로 나눌 수 있다.²³⁾ 이들 기술은 서로 보완과 경쟁 관계를 유지하고 있는데 국내에서는 OWL 기반의 온톨로지로 연구가 집중되는 경향이 있다.²⁴⁾

원래 온톨로지는 철학의 분야 중 하나로, 우리말로로는 보통 ‘존재론’이라고 하고, 때때로 ‘형이상학(metaphysics)’과 동의어로 취급하기도 한다. 학문으로서의 온톨로지는 아리스토텔레스로부터 비롯되었다고 보는 것이 정설이지만, 전산학 혹은 인공지능 분야에서 ‘온톨로지’란 용어가 사용된 것은 1970년대 후반 내지 1980년대 초반부터이다.²⁵⁾ 그루버(Gruber)는 온

20) 한국문화정보센터, 「문화정보 관리·유통 체계를 위한 지식맵 서비스 연구」, 2013, 50쪽.

21) 김현 외, 앞의 책(2016), 163쪽.

22) International Organization for Standardization, <https://www.iso.org/home.html>

23) 오삼균, 「온톨로지 언어의 비교 연구 : W3C OWL과 ISO 토픽맵을 중심으로」,

『한국비블리아학회지』, 제15권, 제2호, 2004, 72쪽.

24) 오삼균, 위의 논문(2004), 72쪽.

25) 한국전산원, 「웹 온톨로지 개발지침 연구」, 2004, 2쪽.

톨로지를 “공유하는 개념화의 형식적이고 명확한 명세”²⁶⁾라고 정의했고, 이 정의는 비교적 널리 받아들여진다. 여기서 ‘공유’란 말은 합의된 지식(concensual knowledge)을 표현해야 한다는 것이고, 개념화(conceptualization)란 특정 영역 또는 분야의 현실 세계와 관련된 개념을 나타내는 추상 모델을 일컫는다. 형식적(formal)이란 온톨로지 내용을 컴퓨터가 읽을 수 있고 처리가 가능한 형태로 표현해야 한다는 것을 의미한다. 명확한(explicit)은 특정 영역을 모델링할 때 사용하는 개념들과 이러한 개념들을 사용할 때 적용되는 제약 조건을 명시적으로 정의해야 한다는 의미이다.²⁷⁾ 온톨로지는 단순히 특정 분야를 표현하는 개념들의 의미만을 정의하는 것이 아니라 각 개념이 지닌 고유한 속성, 개념들 간의 관계 및 이들 사이의 제약 조건, 지식 추론을 위한 공리 규칙과 각 개념 간의 인스턴스를 총체적으로 정의함으로써 그 분야의 지식 체계를 컴퓨터가 해석하고 이해하여 처리할 수 있도록 형식화한 표준 명세서인 것이다.

개념이 무엇인지 알아보자. 개념은 어떤 사물에 대한 일반적이고 본질적인 지식을 말하며, 반드시 단어를 통해서 표현되며, 일반적으로 특정 언어나 단어에 종속되어 있지 않다. 개념은 內包(intension)와 外延(extension)이라는 두 가지 속성을 가진다. 內包는 어떤 개념이 적용되는 대상이 가진 공통적인 성질을 말하며, 外延이란 어떤 개념이 적용되는 대상의 집합을 말한다.²⁸⁾ 예를 들어 皇帝라는 개념에서 內包는 중국의 秦始皇帝가 처음 사용한 호칭으로 ‘나라의 최고 통치자에 대한 칭호’라고 한다면, 外延은 秦始皇帝, 漢高祖 등이 된다. 江原道 북부에 있는 金剛山을 예를 들면 金剛山은 봄에는 金剛山이라고 부르고, 여름에는 蓬萊山, 가을에는 楓嶽山, 겨울에는 皆骨山이라고 부른다. 즉 이들 명칭은 모두 金剛山을 지칭하면서도 각각이 서로 다른 의미를 가지고 있다. 반면에 하나의 구체적인 실체를 지칭하기 때문에 그 外延은 같다고 할 수 있다.

온톨로지를 구축하기 위해서는 적절한 언어를 선택해야 한다. W3C를 중심으로 한 시맨틱 웹 기술의 온톨로지 언어는 크게 RDF(Resource Description Framework), RDFS(RDF Schema), DAML+OIL, OWL 등이 있다. 이들 언어 중 가장 기초가 되는 것은 RDF로 이는 특정 자원에 대한 메타데이터를 기술하는 XML 기반의 프레임워크이다. RDF는 레코드(record)를 하나의 기술 단위로 취급해온 기존의 방식과는 달리 자원, 속성, 속성 값을 하나의 단위로 취급하는 이른바 RDF Triple이 핵심이다. 시맨틱 웹에 사용되는 전반적인 구조는 아래의 시맨틱 웹 아키텍처를 통해서 쉽게 이해할 수 있다.

26) Gruber, T.R. 「A translation approach to portable ontologies」, Knowledge Acquisition, 5(2), 1993, 199~220쪽.

27) 노상규·박진수, 『인터넷 진화의 열쇠 온톨로지』, 가즈토이, 2009, 138쪽.

28) 齋藤孝(저)/최석두·한상길(역), 『온톨로지 알고리즘』1, 한울아카데미, 2008, 118쪽.

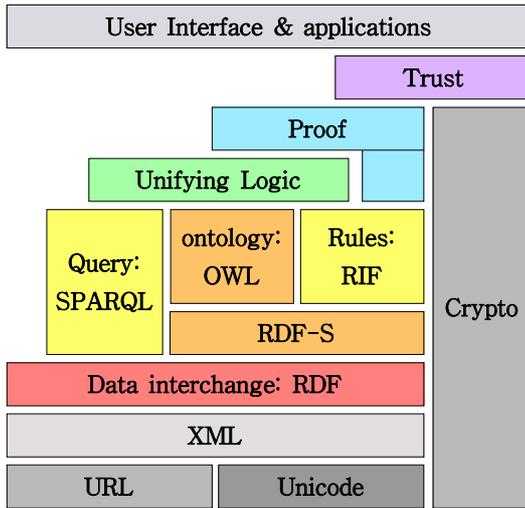


그림 2. 시맨틱 웹 아키텍처

다음으로 온톨로지의 핵심적인 구성 요소는 아래 표와 같이 클래스(Class), 인디비주얼(Individual), 관계(Relation), 속성(Property), 제약조건(Constraint), 공리(Axiom) 등으로 볼 수 있다.

표 1. 온톨로지 구성 요소²⁹⁾

온톨로지 구성 요소	용도	OWL
Class(클래스)	공통의 속성을 가진 개체들을 묶는 범주 a group of individuals that belong together because they share some properties.	owl:Class
Individual(개체)	클래스에 속하는 개체, 각 클래스의 실제 값 Instances of classes	owl:NamedIndividual
Relation(관계)	(같거나 다른 클래스에 속하는) 개체들 사이의 관계 relationships between pairs of individuals	owl:ObjectProperty
Attribute(속성)	개체가 속성으로 갖는 데이터 값 relationships from individuals to data values	owl:DatatypeProperty
Relation Attribute (관계 속성)	관계 정보에 부수되는 속성 attributes related to relations	N/A in OWL
Domain(정의역)	특정 ObjectProperty 또는 DatatypeProperty의 주어가 될 수 있는 클래스를 한정 A domain of a property which limits the individuals to wh	rdfs:domain

29) 인문정보학 온톨로지 설계 가이드라인,

http://kadhlab102.com/wiki/index.php/인문정보학_온톨로지_설계_가이드라인

	ich the property can be applied	
Range(치역)	특정 ObjectProperty의 목적어가 될 수 있는 클래스를 한정 The range of a property limits the individuals that the property may have as its value	rdfs:range

나. RDF(Resource Description Framework)

1) RDF 트리플(Triples)

온톨로지를 구축하기 위한 언어는 여러 가지가 있지만, 그 중 가장 기본적인 언어는 바로 RDF(Resource Description Framework)이다. RDF는 웹 상의 자원(Resource)을 명확하고 논리적으로 표현하는 데이터 모델이며, 이를 기술하기 위한 언어 체계를 말한다. RDF는 사람이 이해할 수 있을 뿐만 아니라 컴퓨터가 쉽게 처리할 수 있도록 논리적인 관계를 주어 (Subject)-술어(Predicate)-목적어(Object)로 표현한 논리적인 모델이다. 이러한 3요소가 하나의 문장(Statement)을 구성하며 이 때문에 RDF 트리플(Triples)이라고 한다. RDF 트리플은 두 자원 간의 관계를 표현한다. RDF에서는 자원의 개념을 웹에서 식별할 수 있는 것이라고 일반화하고 있다. 여기서 자원은 URI(Uniform Resource Identifiers)³⁰⁾로 표현할 수 있는 모든 것을 말하며 사람이나 동물, 식물 등 세상에 존재하는 개체뿐만 아니라 사랑, 취미, 기호 등과 같은 개념까지도 모두 포함한다. 자원과 자원의 관계는 주어에서 목적어로 방향을 가지며 주어와 목적어 간의 관계 즉 서술어를 속성 혹은 프로퍼티(Property)³¹⁾라고 부르기도 한다. RDF는 이들 관계를 그래프³²⁾로 표현할 수 있다. RDF 그래프는 RDF 트리플의 집합이다.³³⁾ 아래와 같은 단독 트리플은 요소 1개로 구성된 집합의 그래프이다.

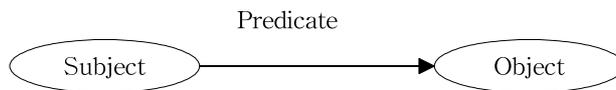


그림 3. rdf 그래프의 예

30) 2013년 RDF1.1은 URI를 IRIs(Internationalized Resource Identifiers)로 변경하였다. 하지만 이들의 차이는 크지 않다. 본 논문에서는 같은 의미로 사용하였다.

31) 프로퍼티(Property)라는 용어는 주로 클래스(Class)와 함께 사용되며, 자원 사이의 관계보다는 주어 자원의 성질을 표현한다는 뉘앙스가 강하다.

32) edge(간선)에 방향을 지정하기 때문에 유향그래프(directed graph)라고도 한다.

33) 神崎正英(지)/황성익·양해술(역), 『시맨틱 웹을 위한 RDF/OWL 입문』, 홍릉과학출판사, 2008, 205쪽.

주어와 목적어는 타원(노드, node)으로 표현하며, 노드와 노드 사이를 연결하는 서술어³⁴⁾는 화살표(arc)³⁵⁾로 표현한다. 예를 들어 ‘蔣玄暉가 昭帝를 시해하다’는 문장은 아래와 같은 RDF 그래프로 표현할 수 있다.

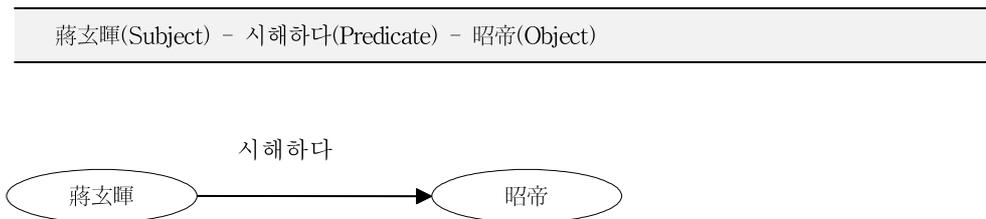


그림 4. 단순 RDF 트리플 구조

목적어는 노드뿐만 아니라 리터럴(문자열)이 올 수도 있다. 이때 그래프에서는 사각형으로 표현하며 더 이상 그래프를 연결해 갈 수가 없다. 이때 리터럴은 ‘plain literal’과 ‘typed literal’ 등 2종류가 있다.

그래프 상에서 같은 식별자를 갖는 노드는 병합시켜서 트리플을 연결시킬 수 있다.³⁶⁾ RDF의 그래프는 테이블 및 트리뿐만 아니라 대규모의 복잡한 데이터도 표현할 수 있으며, 만약 이러한 트리플들은 노드가 같다는 사실을 식별할 수 있으면 상이한 문서에 기술되어 있더라도 병합시킬 수 있다.³⁷⁾ 여기서 핵심은 ‘식별할 수 있다’는 것인데 이 문제는 URI를 사용해서 해결한다. URI는 웹에 존재하는 자원에 대해 전세계적으로 유일한 정체성을 부여한다. 이때 URI는 내부에 공백을 포함하지 않아야 한다는 것이 웹의 관례이다. 예를 들어 ‘part of’와 같은 형태는 ‘CamelCase’³⁸⁾ 규약에 따라 ‘partOf’라는 형태를 사용해야만 한다.

2) 노드(node), 속성(property)

소셜 네트워크 분석에서는 사회 관계망을 분석할 때 보통 그래프 형태로 표현한다. 소셜 네트워크는 하나 이상의 관계(relation) 유형에 의해 연결된 네트워크 구성원의 집합, 즉 액터(actor)의 집합으로 정의될 수 있다.³⁹⁾ 이때 액터는 노드(node, point, vertex)라고도 하며,

34) 프로퍼티(property, 속성), 관계(relation) 등으로도 표현한다.

35) Arc는 predicates 또는 relationships이라고 한다.

36) Protégé에서 인스턴스를 등록할 때 인스턴스의 NamedIndividual의 rdf:about으로 표현되는 ID가 같은 트리플은 자동으로 인스턴스를 병합하여 하나의 RDF 트리플로 만들어 준다.

37) 神崎正英(지)/황성익·양해술(역), 앞의 책(2008), 18쪽.

38) 복수로 구성된 단어에서 공백을 지우고 각 단어의 첫글자는 대문자로 표기하되, 맨 처음 문자는 소문자로 표기하는 것을 말한다. 예를 들어 camelCase, partOf와 같은 방식을 말한다.

39) 광기영, 『소셜네트워크분석』, 청람, 2014, 6쪽.

관계는 라인(line, arc, edge)이라고도 한다. 온톨로지에서도 이러한 용어를 거의 유사하게 사용하기도 하나 일반적으로 노드와 속성 혹은 프로퍼티(Property)라는 용어를 주로 사용한다.

표 2. 온톨로지 노드와 속성 관련 용어

구분	시작	연결	끝
소셜 네트워크	node, point, vertex	line, arc, edge	node, point, vertex
온톨로지	Subject(주어)	Predicate(서술어)	Object(목적어)
	Subject(주어)	Property(속성)	속성값
	node(노드)	Property(속성)	node(노드)
	Resource(자원)	Property(속성), Relation(관계)	Resource(자원), Literal(문자열)
	entity(개체)	Property(속성)	entity(개체)
	domain(정의역)	Property(속성)	range(치역, 공역)
	Individual(instance) ⁴⁰⁾	ObjectProperty DatatypeProperty	Individual(instance) Literal(문자열)
URI	URI	URI, Literal	

주어 자리에 올 수 있는 것은 클래스이며, 클래스는 존재하는 자원 중에서 우리가 관리하고자 선택한 모델에 의해 제한된 정보이다. 주어의 위치에 클래스만 올 수 있다는 것은 리터럴로 표시하는 것은 주어로 올 수 없다는 것을 말한다. 여러 개의 RDF 트리플 중 어느 하나에서 목적어가 다른 트리플의 주어로 사용되었다면 이를 병합할 수 있다. 이때 조건은 이들이 같은 것이라는 식별을 할 수 있어야 한다. 주로 URI 참조⁴¹⁾가 사용된다. 이때 자원을 표현하는 URI 참조는 사람들이 이해하기 쉽게 하기 위해 계층화시키지만 서로 간에는 의미적으로 아무런 연관 관계가 없다. 주어 위치에 올 수 있는 것은 반드시 자원이 되어야 하는 것은 아니고 공백 노드도 올 수 있다. 공백 노드는 단순히 개체와 개체를 연결해 주는 역할을 하며 공백 노드 식별을 위해 rdf:nodeID라는 속성을 사용한다. 이 'rdf:nodeID'는 'rdf:about' 혹은 'rdf:resource' 속성 대신에 사용하며, 외부에서는 참조할 수 없어도 RDF 문서 내에서는 서로 참조할 수 있다. 즉 RDF 구문 내에서 공백 노드를 사용하여 ID를 부여하고, 그 부여된 ID를 주어나 목적어로 참조할 수 있다는 의미이다.

40) 'Individual'은 'Instances'라고도 한다.

41) 'URI 참조(URIfref)'는 URI(Uniform Resource Identifiers)와 플래그먼트 식별자(fragment identifier)를 조합한 것을 말한다. RFC2396에 'URI-reference = [absoluteURI | relativeURI] ["#" fragment]'로 정의하였다.(<https://www.ietf.org/rfc/rfc2396.txt>)

3) URI 참조(URIref)와 자원(Resource)

URI(Uniform Resource Identifiers)로 표현할 수 있는 모든 것을 자원(Resource)라 한다. 여기서 자원은 웹에서 식별할 수 있는 것을 말하며 네트워크 상에 있는 콘텐츠의 메타데이터뿐만 아니라 사람들, 서적, 이벤트 등과 같은 현실 세계의 개체들이나 취미, 기호, 신뢰도 등과 같은 사항들까지도 RDF로 표현이 가능하다.⁴²⁾

팀 버너스 리는 링크드 데이터의 4원칙을 제시했는데 그 중에서 첫 번째 조건으로 든 것이 바로 URI를 이용하여 자원을 식별해야 한다는 것이다. URI를 부여할 때 하나의 개체에 대해 무조건 하나의 URI를 부여해야 한다는 것을 의미하는 것은 아니다. 링크드 데이터에서의 식별은 웹 상에서 고유하게 식별할 수 있어야 한다는 의미이다. 하나의 개체에 유일한 식별자를 부여하는 것은 현실적으로 불가능하고 그럴 필요도 없다.

예를 들어 ‘通鑑節要’에 대해 URI를 부여했으면 다른 기관 또는 다른 사람도 ‘通鑑節要’에 관한 URI를 만들 수 있다. 다만 서로 간의 충돌을 피하기 위해 Namespace를 사용하여 상호 간 의미의 충돌을 막는다. 그러나 가능하면 내부적으로 고유성을 가질 수 있는 식별자를 부여하는 것이 중요하다. 링크드 데이터에서 URI가 온톨로지와 다른 점은 온톨로지는 자원을 식별하는 용도로만 사용되지만 링크드 데이터에서의 URI는 실제로 웹 서버(HTTP 프로토콜)를 통해 발행되어 있어야 한다는 것이다.

RDF에서 URI 참조는 해당 문자열 전체가 하나의 이름이 되며, 한 개의 자원을 표현한다. 즉 HTML을 해석하는 브라우저는 URL 주소에서 URL 본체 부분이 자원을 나타내며, 프래그먼트 식별자(fragment identifier)⁴³⁾는 해당 문서 내의 특정 부분을 가리킨다. 그러나 RDF에서의 URI 참조는 본체와 프래그먼트 식별자에 의한 역할 분담이 없다.⁴⁴⁾ 예를 들어 아래 박스의 URI를 HTML 브라우저는 ‘http://dh.aks.ac.kr/ontologies/historyevent/kno/ontology.index.html’이라는 문서 속에 ‘section3’이라는 이름을 갖는 것으로 해석하나 RDF에서는 전체 주소 ‘http://dh.aks.ac.kr/ontologies/historyevent/kno/ontology.index.html#section3’를 하나의 자원으로 식별한다. 이때 URI는 네트워크 상에 존재하지 않아도 상관없다. 그러나 URL은 실제 웹 자원이 있는 구체적인 위치를 나타낸다.

42) 神崎正英(지)/황성익·양해술(역), 앞의 책(2008), 21쪽.

43) 프래그먼트 식별자는 단편 식별자로 번역하기도 하며 문자 그대로 문서 안의 ‘단편(fragment)’을 의미하며 #으로 시작한다.

44) 神崎正英(지)/황성익·양해술(역), 앞의 책(2008), 13쪽.

"http://dh.aks.ac.kr/ontologies/historicevent/kno/ontology.index.html#section3"

URI

프래그먼트 식별자

이 URI 참조에 의한 식별은 대부분 한 개의 어휘 그룹으로 묶기 위해 공통적인 URL 주소를 갖도록 이름을 부여할 수는 있지만 그렇다고 해서 URL 주소 간에 어떠한 의미적인 공통점이 생기는 것은 아니다.⁴⁵⁾ 즉 아래의 박스에서 'China'과 'QhinDynasty' 사이에는 아무런 의미적 연관 관계가 없다는 말이다. 단지 사람들이 편의상 이해하기 쉽게 하기 위해서 의미를 부여한 것뿐이다.

"http://dh.aks.ac.kr/ontologies/historicevent/kno#/China"

"http://dh.aks.ac.kr/ontologies/historicevent/kno#/China/ChinDynasty"

RDF에서 URI 참조는 RFC2396⁴⁶⁾에서는 ASCII(American Standard Code for Information Interchange) 문자만 사용할 수 있기 때문에 한글과 같은 문자는 URI 참조에 사용할 수 없게 되어 있다. 그래서 ASCII가 아닌 문자를 표현할 수 있도록 찾은 방법이 퍼센트 인코딩(percent-encoding)⁴⁷⁾이다. 퍼센트 인코딩은 어떤 문자열을 인코딩할 때 % 뒤에 그 바이트의 16진수 값을 표기하는 방식이다. 예를 들어 '項羽'라는 어휘는 유니코드의 코드는 '98057FBD'가 되며 이를 UTF-8로 저장하면 'E9A085E7BEED'가 된다. 이를 다시 퍼센트 인코딩으로 바꾸면 '%e9%a0%85%e7%be%bd'가 된다. 이때 주의할 점은 utf-8과 euc-kr 등의 코드에 따라 퍼센트 인코딩은 달라진다. 퍼센트 인코딩을 통해서 ASCII 문자가 아닌 문자를 표현할 수 있게 되었다 하더라도 결국에서는 ASCII를 사용하고 있음에는 변함이 없다.⁴⁸⁾

4) 리터럴(literal)

RDF에서는 목적어(속성값)는 URI에 의해 참조된 자원뿐만 아니라 리터럴이나 수치 등으로 직접 표현할 수 있다. URI 참조는 어떤 자원을 명명하여 간접적으로 표현하는데 비해, 리터럴은 데이터를 직접 표현한다. RDF 그래프에서 목적어를 리터럴로 표현할 수는 있지

45) 神崎正英(지)/황성익·양해술(역), 앞의 책(2008), 22쪽.

46) Network Working Group, Uniform Resource Identifiers (URI): Generic Syntax,
<http://www.rfc-base.org/txt/rfc-2396.txt>

47) RFC 3986에서 퍼센트 인코딩이라는 용어로 정의하였다.

48) 伊藤健太郎·佐藤勇紀·濱崎俊(지)/정재은·이인근·황도삼(역), 『시맨틱 웹』, 두양사, 2015, 113쪽.

만, 주어 또는 서술어를 리터럴로 표현할 수는 없다. 목적어가 리터럴인 그래프는 다른 자원과 연결해 나갈 수 없다. 그런 이유로 RDF 그래프에서 자원은 원으로 표현하는데 비해 리터럴은 사각형으로 표현한다. 리터럴의 표현은 온톨로지를 설계할 때 데이터 타입 프로퍼티에서 데이터 형식을 지정해 주어야 하며, 정확한 데이터 형식의 지정은 정교한 온톨로지 모델을 만들 수 있다는 점에서 중요하다.

OWL에서 리터럴은 RDF 스키마 클래스인 'rdfs:Literal'의 인스턴스이다. 리터럴은 데이터 형식이 없거나 형식화된 것 모두일 수 있다. RDF/XML에서 리터럴의 형식은 rdf:datatype 속성으로 표현한다.⁴⁹⁾

```
<datePublished rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2015-08-17T19:02:41</datePublished>
<hlod:careerName RDF:datatype="http://www.w3.org/2001/XMLSchema#string">황후궁대부(皇后宮大夫)</hlod:careerName>
```

RDF 그래프의 리터럴의 데이터 형식 중 형식이 없는 일반 문자열 방식은 RFC-3066⁵⁰⁾에서 정의한 언어 태그를 사용한다. 주로 요소의 내용이 어떤 언어로 기술되어 있는지를 xml:lang 속성에 속성값으로 표시한다.⁵¹⁾ 한국어의 경우 'kr', 영어는 'en', 중국어는 'zh', 일본어는 'ja'로 지정한다. 다음으로 형식화된 문자열은 XML 스키마⁵²⁾에 정의된 데이터 유형을 사용한다.⁵³⁾ 다시 정리하면 리터럴 표현은 RDF 표준에 포함된 'rdfs:Literal'과 'rdf:XMLLiteral'과 형식이 없는 일반 문자열 방식과 형식화된 문자열을 사용할 수 있다.

49) 이 데이터형식에 대한 자세한 내용은, RDF 개념 문서(RDF Concepts)를 참고할 수 있다.

50) 「Using Language Identifiers(RFC 3066)」, <https://www.ietf.org/rfc/rfc3066.txt>

51) 伊藤健太郎 외(저)/정재은 외(역), 앞의 책(2015), 100쪽.

52) 「XML Schema Part 2: Datatypes」, W3C,

<https://www.w3.org/TR/2001/REC-xmlschema-2-20010502>

53) 「Resource Description Framework (RDF): Concepts and Abstract Syntax」, W3C,

<https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#ref-xml-schema2>

표 3. XML 스키마에서 정의된 데이터 형식

데이터 형식	불리언 형식	숫자 형식	시간 형식 ⁵⁴⁾	바이너리 형식
xsd:string	xsd:boolean	xsd:decimal	xsd:dateTime	xsd:hexBinary
xsd:normalizedString		xsd:float	xsd:time	xsd:base64Binary
xsd:token		xsd:double	xsd:date	xsd:anyURI
xsd:language		xsd:positiveInteger	xsd:gYearMonth	
xsd:NMTOKEN		xsd:integer	xsd:gYear	
xsd:Name		xsd:short	xsd:unsignedByte	
xsd:NCName		xsd:unsignedShort	xsd:gMonthDay	
		xsd:nonPositiveInteger	xsd:gDay	
		xsd:nonNegativeInteger	xsd:gMonth	
		xsd:negativeInteger		
		xsd:long		
		xsd:int		
		xsd:byte		
		xsd:unsignedLong		
		xsd:unsignedInt		

온톨로지 설계에서 Datatype Property를 설정할 때 가장 많은 고민을 하게 만드는 것은 아마도 리터럴을 어떤 형식과 숫자 형식의 표현일 것이다. 리터럴은 주로 language⁵⁵⁾, string, Literal, XMLLiteral을 사용하며, 숫자에는 주로 int, float, double을 사용한다. 숫자 형식은 정수형과 실수형으로 구분할 수 있는데 정수형은 int, long, short를 사용하며, 실수형은 float, double을 사용한다. 본 논문에서는 형제 순서 등은 정수형인 int를 사용하였으며, 지명의 경·위도 좌표값은 float를 사용하였다.

5) 공백 노드(Blank node)

RDF 그래프는 이름을 갖지 않는 공백 노드(Blank node)를 이용하여 구조적인 데이터 표현이 가능하다. 공백 노드는 자원이 알려지 있지 않지만 속성(관계)은 알려져 있는 것에 사용한다. 공백 노드는 외부로부터 참조할 수는 없지만 자원의 주어 혹은 목적어가 될 수 있

54) XML Schema Part 2: Datatypes Second Edition, W3C,

<https://www.w3.org/TR/xmlschema-2/#date>

55) XML Schema 1.0에서 xsd:language은 데이터 형식을 정규식

'[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*'으로 정의하였다. 이 조건을 충족시키는 예로 'ko(Korean), en(English), en-GB(UK English), en-US(US English), fr(French), de(German), es(Spanish), it(Italian), nl(Dutch), zh(Chinese), ja(Japanese)' 등이 있다.

다. 예를 들어 ‘韓廣이 스스로 서서 燕王이 되었다’라는 문장을 RDF 그래프로 표현하면 ‘韓廣이 자립하다’와 ‘韓廣이 燕王이 되다’라는 두 개의 문장으로 표현이 가능하다. 이 두 개의 문장은 자원의 식별이 가능하면 하나로 병합할 수 있다. 이때 ‘자립하다’는 자동사로 목적어를 필요로 하지 않는다. 이런 경우에 공백 노드를 사용하여 RDF 그래프로 표현할 수 있다.

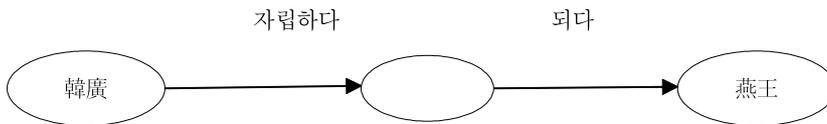


그림 5. 공백 노드의 예시

공백 노드는 단순히 주어와 목적어를 중개해 주는 역할을 하며, 그 값이 무엇인지를 언급하지 않고 자원을 표현할 수 있다. 공백 노드를 주어로 하면 여러 개의 속성을 연결한 RDF 그래프를 그릴 수 있다. 즉 단순한 주어-술어-목적어 관계에서 하나의 주어를 중심으로 복수의 술어, 목적어를 연결할 수 있는 것이다. 예를 들어 아래와 같이 논문 ‘전근대인물 식별을 위한 인명 데이터베이스 연구’의 저자는 성명이 ‘윤종웅’이고, 이메일 주소가 ‘yjw2999@gmail.com’이라는 것을 양자 관계를 통해서 표현하기 어렵다. 이러한 경우에 공백 노드를 사용하여 논문의 저자의 이름이 ‘윤종웅’이고 이메일 주소가 ‘yjw2999@gmail.com’이라는 것을 표현할 수 있다.

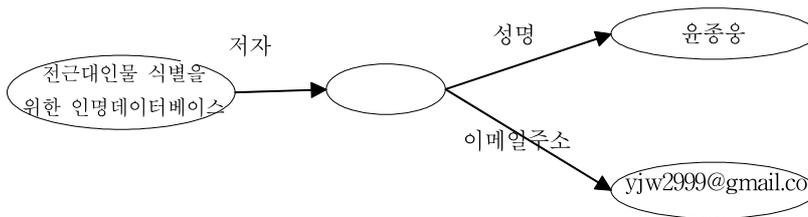


그림 6. 공백 노드를 이용한 복수의 속성 표현

6) 구체화(Refication)

RDF 트리플은 정보를 주어-서술어-목적어를 하나의 단위로 하여 표현한다. 그러나 이러한 이진 관계로는 다양한 정보를 표현하는데 한계가 있다. 이러한 한계를 극복하기 위해 RDF에서 제공하는 방법이 구체화이다. 예를 들어 하나의 RDF의 트리플이 때로는 또 다른 서술문의 목적어가 되어야 하는 경우가 있다. 어떤 문장이 서술하고 있는 사항을 기술 대상으로 해야 할 경우에는 하나의 서술문에 대한 또 다른 서술문을 추가하여 모델링 하는 것

을 구체화(Reification)라고 한다. 예를 들어 “柳渾이 말하기를 ‘吐藩은 豺狼과 같습니다.’”라는 문장에서 ‘柳渾’은 주어부, ‘말하기를’은 서술부, ‘吐藩은 豺狼과 같습니다.’는 전체 서술문이 목적부가 된다. 이때 RDF에서 문장의 구체화는 노드가 ‘문장’임을 나타내기 위해서 ‘rdf:Statement’ 클래스를 사용하고, 주어와 술어 그리고 목적어를 나타내기 위해서, 각각 ‘rdf:subject(주어), rdf:predicate(술어), rdf:object(목적어)’와 같은 속성을 사용한다. 이와 같은 구체화 그래프와 원래의 문장 사이의 관계를 나타내면 다음과 같다.

柳渾이 말하기를 “吐藩은 豺狼과 같습니다.

1. 柳渾이 #1 말하다
2. #1 吐藩은 豺狼과 같다

<pre><rdf:Description rdf:about="柳渾"> <kno:말하다> <rdf:Description rdf:about="#1"/> </kno:말하다> </rdf:Description></pre>	<pre><rdf:Description rdf:about="吐藩"> <kno:같다> <rdf:Description rdf:about="豺狼"/> </kno:같다> </rdf:Description></pre>
---	---

이 두 문장을 합쳐서 구체화하면 다음과 같다.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY kno "http://dh.aks.ac.kr/ontologies/historicevent/kno#">
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
]>
<rdf:RDF xmlns:kno="http://dh.aks.ac.kr/ontologies/historicevent/kno#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Statement>
    <kno:말하다 rdf:resource="柳渾"/>
    <rdf:subject rdf:resource="吐藩"/>
    <rdf:predicate rdf:resource="같다"/>
    <rdf:object rdf:resource="豺狼"/>
  </rdf:Statement>
</rdf:RDF>
```

구체화에 필요한 요소를 정리하면 다음과 같다.

표 4. 구체화에 필요한 클래스

rdf:Statement	RDF 트리플로부터 만들어지는 문장(Statement)의 모델을 나타내는 클래스
rdf:subject	RDF의 문장을 나타내는 rdf:Statement 타입의 자원과 해당 문장의 주어가 되고 있는 자원을 연결하는 속성
rdf:predicate	rdf:Statement 타입의 자원과 해당 문장의 술어가 되고 있는 자원을 연결하는 속성
rdf:object	rdf:Statement 타입의 자원과 해당 문장의 목적어가 되고 있는 자원을 연결하는 속성

Statement 형을 나타내는 트리플과 원래 문장의 주어, 술어, 목적어 각각에 대응하는 트리플의 4요소에 의해 선언문을 구체화한다. 위의 RDF/XML 문장을 RDF 그래프로 표현하면 다음과 같다.

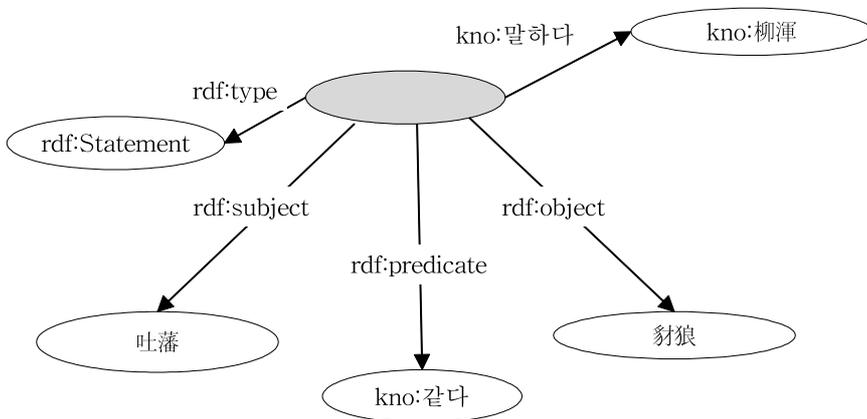


그림 7. 구체화 그래프

다. RDF Schema

RDF를 이용하여 다양한 자원과 자원 간의 관계를 기술하기 위해서는 일정한 어휘가 필요하다. RDF Schema란 자원을 기술할 때 사용할 어휘를 정의해 놓은 것이다.⁵⁶⁾ 사람들은 RDF 트리플을 이용하여 자원을 자유롭게 기술할 수 있다. 그러나 여러 사람들이 개별적으로 RDF 트리플을 작성하다보면 RDF의 구조가 동일한 구조로 되어 있지만은 않을 것이다. 이런 차이를 사전에 방지하기 위해 사용한 방법이 바로 RDF 스키마이다. RDF에는 프로퍼티의 주어에 올 수 있는 자원을 제한하거나 비슷한 자원을 하나로 묶어서 클래스로 표현하

56) 伊藤健太郎 외(지)/정재은 외(역), 앞의 책(2015), 113쪽.

는 기능이 없다. 따라서 자원들이 RDF 트리플로 연결되어 있어도 그것이 실질적으로 어떤 의미를 지니는지 컴퓨터가 알 수 없다. 이러한 문제를 해결하기 위해 RDF Schema에는 RDF에 없는 클래스(Class)라는 개념을 도입하였다. 클래스는 동일한 속성을 지니고 있어 하나의 분류로 모아지는 개체들의 그룹을 의미한다.⁵⁷⁾ 또한 일정한 同質性을 가지고 있는 집합을 의미한다. 예를 들어 『通鑑節要』, 『資治通鑑』, 『資治通鑑綱目』이 모두 전통시대에 간행된 역사서라는 공통점을 가지고 있기 때문에, 이를 묶어서 ‘역사서’라는 클래스를 만들 수 있다. 그러나 편찬 체제로 클래스를 설정한다면 앞의 두 책은 ‘편년체’라는 클래스가 될 것이며, 『資治通鑑綱目』은 ‘강목체’가 될 것이다. 이러한 클래스는 그 의미에 따라 ‘rdfs:subClassOf’라는 어휘를 통해 상위 클래스와 연결되면서 계층을 형성한다. 하위 클래스⁵⁸⁾에 속한 인스턴스는 자동으로 상위 클래스에도 속하게 된다. 이러한 클래스는 RDF 스키마 자원인 ‘rdfs:Class’와 ‘rdfs:Resource’, 그리고 ‘rdf:type’와 ‘rdf:subClassOf’ 속성을 사용하여 기술한다.

RDF 스키마에서 클래스를 정의(표현)하기 위해서는 자원이 ‘rdfs:Class’ 클래스에 속한다는 것을 ‘rdf:type’ 속성으로 표시한다. 예를들면 ‘knoDynasty:後周’이라는 이름을 갖는 자원은 Class의 인스턴스 중의 하나라는 것을 ‘knoDynasty:後周 rdf:type rdfs:Class’와 같은 트리플로 표현한다.

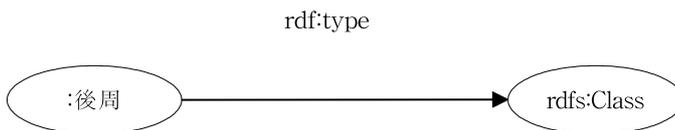


그림 8. ‘rdf:type’을 이용하여 ‘rdfs:Class’에 속함을 명시

RDF 스키마는 주어, 목적어가 되는 자원에 대해서뿐만 아니라 서술어인 프로퍼티가 되는 자원에 대해서도 정의한다.

57) 노상규·박진수, 앞의 책(2009), 131쪽.

58) 하위 클래스는 subClass를 말하며, 서브 클래스, 하위 클래스 등으로 혼용해서 사용되고 있다.

이러한 예로 서브 프로퍼티(subProperty), 슈퍼 프로퍼티(superProperty) 등이 있다. 본 논문에서는 서브 클래스(subClass), 슈퍼 클래스(superClass) 등의 용어를 모두 상위, 하위 등으로 표현하였다.

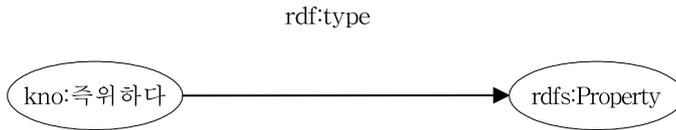


그림 9. 'rdf:type'을 이용하여 rdfs:Property에 속함을 정의

RDF에서 프로퍼티가 자원과 자원의 관계를 표현한다면 RDF Schema에서는 클래스와 클래스 간의 관계를 표현한다. RDF는 어떤 프로퍼티의 주어와 목적어 위치에 올 수 있는 개체를 지정하지 않았는데 RDF Schema는 특정 프로퍼티의 주어 위치에 올 수 있는 클래스와 목적어 위치에 올 수 있는 클래스 혹은 리터럴을 명시적으로 지정할 수 있게 하였다. 이를 定義域(domain)과 值域(range)⁵⁹⁾이라고 하는데 定義域은 RDF 트리플에서 주어(subject)이며 프로퍼티는 서술어(predicate), 值域은 목적어(Object)이다. RDF 스키마에서 클래스와 프로퍼티의 관계는 아래 그림을 통해서 명확하게 이해할 수 있다. RDF 스키마는 모든 것들이 'rdfs:Resource'의 하위 클래스이며, 프로퍼티는 RDF의 것을 그대로 수용하였다. 아래 그림에서 보면 황토색이 클래스이며 가운데의 노란색이 프로퍼티이다. 모든 프로퍼티는 rdf:Property 클래스는 속하는 것으로 표현된다.

59) RDF 트리플에서 목적어에 해당하는 것을 值域 혹은 共域으로 번역하기도 한다. 본 논문에서는 모두 值域이란 용어를 사용한다.

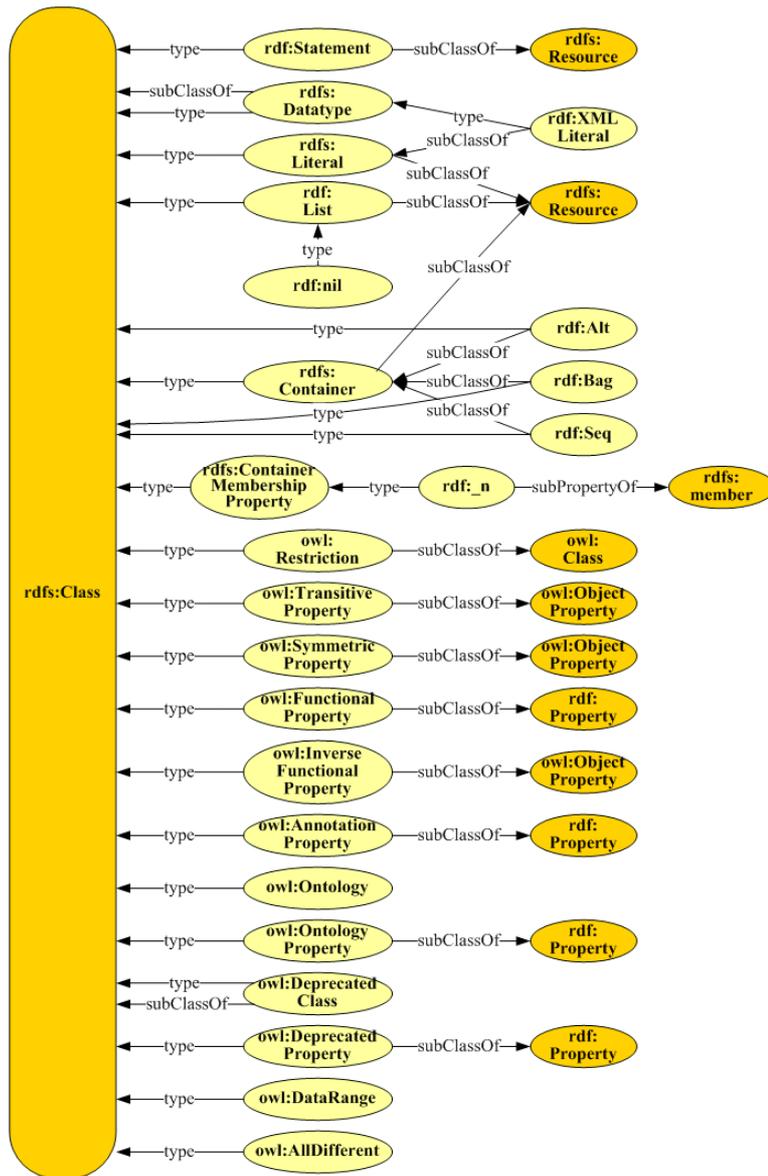


그림 10. RDFS와 OWL의 프로퍼티60)

60) RDFS and OWL Classes,

<https://www.infowebml.ws/rdf-owl/graphical-representations.htm#Statement>

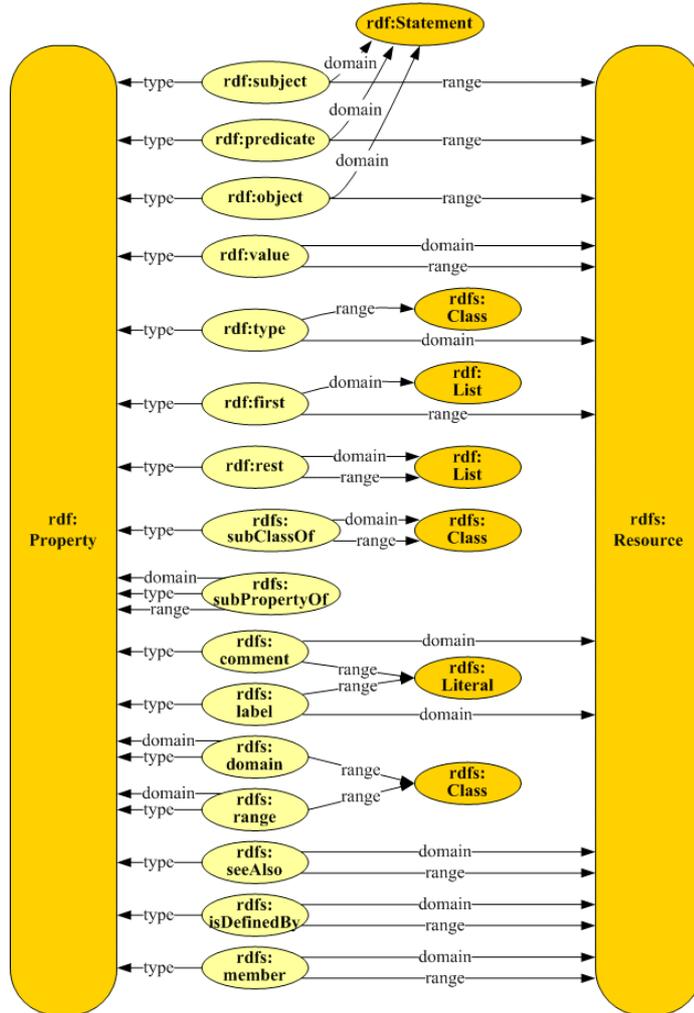


그림 11. RDF와 RDFS의 프로퍼티⁶¹⁾

라. OWL(Web Ontology Language)

OWL(Web Ontology Language)은 웹의 자원을 기술하는 온톨로지 언어로서 2004년에 W3C에 의해서 권고되었다. OWL은 RDF와 RDF Schema의 문제점을 보완하기 위해 개발

61) RDFS and OWL Classes Sta,

<https://www.infowebml.ws/rdf-owl/graphical-representations.htm#Statement>

된 언어이며 현재로서는 다른 온톨로지 언어에 비해 표현력이나 추론 능력에 있어 가장 뛰어난 언어라고 평가받고 있으며 가장 널리 사용되고 있다. OWL은 OWL Lite, OWL DL, OWL Full 등이 있다. OWL Lite는 클래스의 분류 계층과 간단한 제약 사항 표현 기능을 요구하기에 적합한 언어이기 때문에 표현력의 제약이 많다. 그렇기 때문에 유의어 사전이나 간단한 분류 체계 그리고 시소러스를 빠르고 손쉽게 OWL화하기 위한 용도에 적합하다. OWL DL은 기술 논리(Description Logic)⁶²⁾에 기반한 언어이다. OWL DL은 OWL에서 정의한 모든 어휘를 포함하고 있으나 어휘를 사용할 때에는 사전에 정해진 제약 사항을 준수해야 한다. 먼저 클래스는 인스턴스 또는 프로퍼티일 수 없으며, 프로퍼티는 인스턴스 또는 클래스일 수 없고, 프로퍼티는 Object Property이거나 Datatype Property이어야 한다.⁶³⁾ 그리고 inverse of, inverse functional, symmetric, transitive 함수의 특성을 함의한다. 그러나 이 함수들은 Datatype Property에서는 구체화될 수 없다. 또한 OWL DL은 기수성 제약 조건이 추이적 속성이나 역속성 또는 다른 상위 속성으로 선언될 수 없도록 요구한다.⁶⁴⁾

OWL Full은 RDF의 모든 문법을 사용할 수 있으며, 최대의 표현력을 제공하나 아직까지 OWL Full을 지원하는 추론 엔진을 구현하는 것은 매우 어려운 것으로 판단된다.⁶⁵⁾ 이러한 점을 고려하여 본 논문에서는 온톨로지 언어로는 OWL DL를 사용한다.

표 5. OWL 언어 비교⁶⁶⁾

OWL Lite		OWL DL/OWL FULL	
속성 특성	ObjectProperty DatatypeProperty inverseOf TransitiveProperty SymmetricProperty FunctionalProperty InverseFunctionalProperty	클래스 공리	oneOf, dataRange disjointWith equivalentClass rdfs:subClassOf
동치	equivalentClass equivalentProperty sameAs	부울 조합	unionOf complementOf intersectionOf unionOf complementOf intersectionOf

62) 기술논리(Description Logic)는 기본적인 개념을 표현하는 기호(symbol)와 개념들은 연결하는 구성자(constructor)를 사용하여 애플리케이션 도메인의 개념적(conceptual) 또는 용어적 지식(terminological knowledge)을 표현하기 위한 지식 표현 체계이다.

63) 김상균, 『TL-OWL : 시간 지식의 표현과 추론을 지원하는 온톨로지 언어』, 충남대학교 대학원 박사학위논문, 2008, 8쪽.

64) 『OWL 웹 온톨로지 언어 참고서』, 한국정보통신기술협회, 2008, 43쪽.

65) 노상규·박진수, 앞의 책(2009), 138쪽.

66) OWL 웹 온톨로지 언어 개요 W3C 권고안 한글 번역문, W3C, <http://www.w3c.or.kr/Translation/REC-owl-features-20040210>

	differentFrom AllDifferent distinctMembers	관계 차수	minCardinality maxCardinality cardinality
속성 제약	Restriction onProperty allValuesFrom someValuesFrom	속성값 정보	hasValue
관계 차수	minCardinality maxCardinality cardinality		
클래스 교차	intersectionOf		
데이터타입	xsd datatypes		
속성 특성	Ontology imports intersectionOf rdfs:label rdfs:comment rdfs:seeAlso rdfs:isDefinedBy AnnotationProperty OntologyProperty xsd datatypes		

위의 표에서 RDF 및 RDF 스키마에 정의되어 있는 어휘에 대해서는 각각 접두사인 rdf: 및 rdfs:를 달아 두었으며 접두사가 붙어 있지 않은 어휘는 OWL 어휘이다.

1) RDF와 OWL

OWL은 클래스(Class)와 속성(Property)을 통해 의미를 표현한다. 클래스는 동일한 속성을 지니고 있어 하나의 분류로 모아지는 개체들의 그룹을 의미한다.⁶⁷⁾ OWL에서 클래스의 표현은 'owl:Class'로 하며, 오브젝트 프로퍼티는 'owl:ObjectProperty'로 데이터 타입 프로퍼티는 'owl:DatatypeProperty'로 표현한다.

온톨로지는 RDF 그래프로 표현할 수 있으며 그대로 RDF 트리플 세트가 된다. RDF 그래프와 마찬가지로 OWL 온톨로지 그래프는 다양한 구문 형식으로 표현할 수 있다. OWL 온톨로지의 의미는 RDF 그래프에 의해서 결정된다. 따라서 RDF 트리플은 동일한 의미와 결과를 갖는 서로 다른 형태의 RDF/XML 구문이 허용된다. 아래의 두 구문은 완전하게 동일한 의미를 전달한다.⁶⁸⁾

67) 노상규·박진수, 앞의 책(2009), 131쪽.

68) OWL Web Ontology Language Reference,

<https://www.w3.org/TR/owl-ref/#ref-rdf-concepts#ref-rdf-concepts>(W3C에서 2004년 10월에

```
<rdf:Description rdf:about="#Continent">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
```

아래 트리플을 살펴보면 'rdf:type'의 단편 식별자인 'Class'를 클래스명으로 하고, 'rdf:Description' 노드의 'rdf:about' 속성 대신 'rdf:ID'로 표현하였다.

```
<owl:Class rdf:ID="Continent"/>
```

'rdf:about'은 RDF 트리플에서 URI 참조를 할당할 때 사용한다.

아래표는 RDF에서 처음 도입된 개념과 그 후 RDFS와 OWL에서 도입된 개념을 보여주고 있다. 예를 들어 Class라는 개념은 RDFS에서 처음 도입된 개념이다.

표 6. RDF-RDFS-OWL Table of Contents⁶⁹⁾

RDF CONCEPTS & syntaxTerms	RDF Schema CONCEPTS	OWL CONCEPTS
_n	Class	AllDifferent
about	comment	allValuesFrom
Alt	Container	AnnotationProperty
Bag	ContainerMembershipProperty	backwardCompatibleWith
datatype	Datatype	cardinality
Description	domain	Class
first	isDefinedBy	complementOf
ID	label	DataRange
li	Literal	DatatypeProperty
List	member	DeprecatedClass
nil	range	DeprecatedProperty
nodeID	Resource	differentFrom
object	seeAlso	disjointWith
parseType	subClassOf	distinctMembers
parseType="Collection"	subPropertyOf	equivalentClass
parseType="Literal"		equivalentProperty
parseType="Resource"		FunctionalProperty
predicate		hasValue
Property		imports
RDF		incompatibleWith
resource		intersectionOf
rest		InverseFunctionalProperty

권고안으로 제시)

69) W3C, RDF - RDF Schema - OWL references, <https://www.infowebml.ws/rdf-owl/>

Seq Statement subject type value XMLLiteral		inverseOf maxCardinality minCardinality Nothing ObjectProperty oneOf onProperty Ontology OntologyProperty priorVersion Restriction sameAs someValuesFrom SymmetricProperty Thing TransitiveProperty unionOf versionInfo
--	--	--

2) 클래스(Class)와 속성(Property)

OWL에서 속성(Property)은 개체(Instance) 간의 관계를 표현해 준다. 즉, Property는 두 인스턴스를 하나로 연결한다. 이 속성은 다시 ‘오브젝트 프로퍼티(ObjectProperty)’와 ‘데이터 타입 프로퍼티(DatatypeProperty)’로 구분한다. ‘오브젝트 프로퍼티’는 클래스 요소들 간의 관계를 표현하며, ‘데이터 타입 프로퍼티’는 클래스 요소가 취해야 하는 데이터의 형식과 값을 기술한다. 이때 클래스는 URI 참조를 사용하고, 값은 리터럴을 의미한다.

RDF에서 속성이 자원과 자원의 관계를 나타낸 것이라면, RDF Schema에서는 속성이 클래스와 클래스의 관계를 나타내는 것이다. Class 개념을 통해 모든 개체를 효율적으로 묶어 표현할 수 있으며, Class는 다른 Class와 관계를 프로퍼티를 이용하여 다양한 정보를 표현할 수 있다. 이들은 定義域(Domain)과 共域(值域, Range)이 있어 속성의 주어부와 목적부에 쓸 수 있는 클래스를 지정할 수 있다. 클래스의 설계를 위해서는 “우리가 사용하려 하는 두 개 이상의 속성과 두 개 이상의 개체를 지닌 동질성의 의미를 가진 독립적인 집합”이라는 문장을 통해서 어느 정도 접근할 수 있다. 독립적인 집합이라는 것은 어떤 다른 집합과는 본질적으로 구분되는 자기만의 고유한 집합을 말한다. 위에 열거한 특징들은 어떤 집합이 클래스인지 아닌지 판단할 수 있는 중요한 기준이 된다. 클래스에는 반드시 정의가 필요하다. 정의가 필요한 이유는 클래스 정의를 통해서 클래스의 범주에 속하는 것과 그렇지 않은 것을 식별할 수 있기 때문이다. 또 집합을 클래스로 분류하기 위해서는 집합을 구성하고 있는 요소 간에 동질성이라는 공통 분모를 가져야 한다. 이러한 동질성은 온톨로지 모델 명세

서 등에 정확하게 표현해야 한다.⁷⁰⁾ 예를 들어 ‘직원’이라는 클래스를 정의했다면 여기에는 ‘정규직’도 있을 수 있고, ‘비정규직’, ‘촉탁직원’, ‘직원으로서 구체적인 부서를 배정받지 못한 직원’도 있을 수 있다. 만약 직원이라는 클래스 정의를 ‘회사에서 등록된 정규직’으로 한다면 계약직이나 협력업체 직원 등은 동질성을 잃게 된다. 반면 ‘회사에서 일하는 모든 사람’이라 하면 이들은 모두 동질성을 갖게 된다.

클래스의 분류는 리터럴과 연관을 지어서 생각해야 한다. 클래스로 분류한다는 것은 클래스 내의 인스턴스가 다른 클래스의 인스턴스 혹은 리터럴과 관계를 맺는다는 것을 의미한다. 이때 주어와 목적어 관계가 생성이 된다. 목적어가 인스턴스라면 다른 인스턴스와 관계를 맺으면서 주어가 될 수 있다. 이런 식으로 트리플이 계속해서 연결될 수 있는데 리터럴은 그렇지 못하다. 위에서 리터럴은 데이터 자체를 직접 표현하며, RDF 그래프에서 목적어는 될 수 있지만, 주어 또는 술어를 리터럴로 표현할 수는 없다고 하였다. 그렇기 때문에 리터럴은 정보의 종착 지점을 의미한다. 예를 들어 역사서에 나오는 어떤 한 지역과 관련된 지식이 많다면 이 지역을 중심으로 RDF 트리플이 연결이 되는 것이 많을 수 있다. 그렇다면 이는 클래스로 분류하는 것이 좋을 것이다.

마지막으로 클래스의 분류는 목적과 용도에 적합할 정도로만 계층화시켜야 한다. 온톨로지는 분류가 가능한 모든 부분을 계층화시키지 않는다. 우리가 관리하고자 하는 정도까지만 계층화시키면 된다.

3) 속성(Property)의 표현

다음은 프로퍼티의 표현에 대해서 알아본다. 프로퍼티는 크게 Object Property와 DataType Property로 구분할 수 있다. 이들은 OWL DL/Lite에서는 엄격하게 구분이 되지만 OWL Full에서는 명확하게 구분이 되지 않는다. 이들 두 개의 Property는 상위, 하위 프로퍼티로 서로 섞어서 정의할 수 없으며, 동일한 이름을 사용하는 것도 허용하지 않는다.

70) 한국정보화진흥원에서 발간한 『링크드 데이터 구축 공정 가이드 v1.0』을 참조하면 구축 과정에서 필요한 산출물에 대해서 상세하게 알 수 있다.

표 7. Object Property와 DataType Property 비교

구분	Object Property	DataType Property
주어	○	X
목적어	○	○
형식	URI 참조	언어 형식, 정형화된 형식
확장	가능	불가능

아래 문장을 보면 ‘秦始皇帝’와 아들인 ‘胡亥’는 서로 부자 관계이며, ‘秦始皇帝’를 기준으로 ‘胡亥’가 아들이 되며, ‘胡亥’의 관점에서 보면 ‘秦始皇帝’가 아버지가 된다. 이 둘 중 어느 한 형식으로 표현하면 나머지는 유추에 의해 컴퓨터가 이해할 수 있다. 이러한 역관계는 지식-획득이라는 측면에서 보면 이러한 두 가지 정보를 명시적으로 나타내는 것이 좋다.⁷¹⁾

秦始皇帝(Subject) - 아들(Predicate) - 胡亥(Object) 胡亥(Subject) - 아버지(Predicate) - 秦始皇帝(Object)

이러한 관계의 또 하나 유형으로 能動과 被動 관계가 있다. 『通鑑節要』⁷²⁾ 後漢紀의 隱皇帝 漢乾祐三年 기사에 다음과 같은 내용이 있다.

郭威가 마침내 그의 養子인 郭榮을 남겨두어 鄴都에 鎮駐하게 하고, 郭崇威에게 명하여 기병을 거느려 先鋒이 되게 하고 자신은 大軍을 거느리고 그 뒤를 이어서 封丘縣에 이르니, 인심이 흉흉하고 두려워하였다.

漢主(劉承祐)가 慕容彥超 등을 보내어 군대를 거느리고 이들을 막게 하였는데, 慕容彥超 등이 싸움에 패하고 마침내 돌아왔다

이날 漢主가 친히 나가 군사를 위로하다가 군사들이 패하여 亂兵에게 시해당하였다

이 문장에서 보면 ‘漢主’가 ‘慕容彥超’를 시켜서 막게 한 ‘이들’은 ‘郭崇威의 기병’이거나 ‘郭威의 대군’이 될 것이다.

71) 박지영, 『온톨로지 개발을 위한 일반 지침(Ontology Development 101: A Guide to Creating Your First Ontology)』, 도서관 정책자료, 2008, 31쪽.

72) 五代紀 後漢紀 隱皇帝 [庚戌] 漢乾祐三年, 通鑑節要(온라인 참조: 전통종합DB, 전통문화연구원, http://db.cyberseodang.or.kr/front/alphaList/BookMain.do?mId=m02&bnCode=jti_2b0109&titleId=C1026).

劉承祐 - 막게하다 - 이들
 慕容彥超 등 - 패하여 돌아 오다 - (이들에게)
 劉承祐 - 시해당하다 - 亂兵

이 문장에서 ‘劉承祐가 亂兵에게 시해당하다’를 능동형으로 바꾸면 ‘亂兵이 劉承祐를 시해하다’가 된다. 그런데 ‘亂兵’을 만약 리터럴로 처리한다면 리터럴은 주어가 될 수 없기 때문에 문장 그대로 ‘劉承祐’를 주어로 하고, 목적어는 ‘亂兵’으로 하면 서술어를 피동형인 ‘시해당하다’로 표현하면 된다.

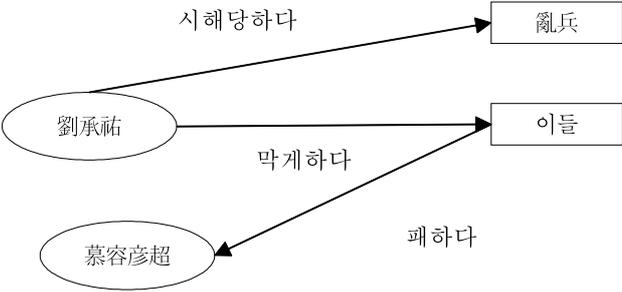


그림 12. RDF 그래프

아래 문장에서는 1은 項羽를 주어로 하고, 義帝를 목적어로 하여 項羽가 義帝를 죽인 것으로 표현하였으며, 2는 義帝를 주어로 하여 項羽를 목적어로 하여 義帝가 項羽에게 죽음을 당한 것으로 표현하였다.

1. 項羽(Subject) - 죽이다(Predicate) - 義帝(Object)
2. 義帝(Subject) - 죽임을 당하다(Predicate) - 項羽(Object)

아래 그래프를 보면 項羽와 義帝의 위치가 서로 바뀌었으나 그 의미는 동일하다.

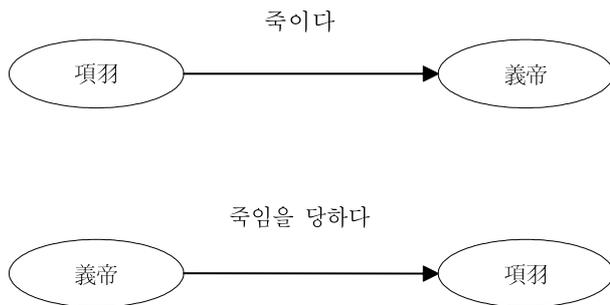


그림 13. 項羽와 義帝의 관계를 표현한 RDF 그래프

4) 공리(Axiom)

온톨로지 설계에서 公理(Axiom)는 ‘가치가 있다고 간주되거나 그 자체로 명백하다’라는 의미를 가지고 있으며 어떤 명제들을 증명하기 위한 전제로 이용되는 가장 기본적인 가정을 가리킨다. 지식이 참된 것이 되기 위해서는 근거가 필요하고 그 근거를 소급해 보면 더 이상 증명하기가 곤란한 명제에 다다르게 되는데 이것이 바로 공리이다.⁷³⁾

가) 클래스 명명

URI 참조에 의한 클래스 식별은 아래와 같이 OWL 클래스 owl:Class에 rdf:about 속성을 사용하고, URI 참조로 식별한다. 클래스의 인스턴스는 개체(Individual), 인스턴스의 집합은 클래스의 外延(Class Extension)이라고 한다.

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Event -->
<owl:Class rdf:about="&kno:Event">
<rdfs:comment>사건 정보</rdfs:comment>
</owl:Class>
  
```

나) 개체의 열거

클래스는 그 클래스에 속한 개체를 열거하여 정의할 수 있다. OWL에서는 ‘owl:oneOf’ 프로퍼티를 사용하여 개체 멤버를 정확하게 열거한다. ‘owl:oneOf’는 사용에 주의해야 하는데 오직 클래스의 멤버가 쉽게 바뀌지 않는 상황 혹은 적어도 자주 바뀌지 않을 때에 사용되어야 한다.⁷⁴⁾ ‘owl:oneOf’는 열거한 개체들이 해당 클래스에 속하는 개체이고 그 이상도 이

73) 강현민, 「공리정의를 이용한 인명접근점제어 온톨로지 기술에 관한 연구」, 『한국문헌정보학회지』, 제46권 제2호, 2012, 159~160쪽.

하도 아님을 명시하기 위해 RDF 컬렉션을 목적으로 한다. ‘owl:oneOf’ 프로퍼티는 OWL Lite에서는 사용할 수 없다. 예를 들어 간지의 경우에 전체 구성원이 60개이며 이는 시간이 지나도 변하지 않는다.

```

<owl:Class rdf:ID="&#x2162;SexagenaryCycle">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="&#x2162;甲子"/>
    <owl:Thing rdf:about="&#x2162;乙丑"/>
    .....
  </owl:oneOf>
</owl:Class>

```

다) 프로퍼티 제약

OWL에서는 그 클래스에 속하는 모든 개체가 만족하는 프로퍼티에 관한 제약을 기술하는 것으로 클래스를 정의할 수 있다.

표 8. 프로퍼티 제약

종류	어휘	내용
값에 의한 제약	owl:allValueFrom	모든 프로퍼티 값은 지정된 클래스의 인스턴스
	owl:someValueFrom	적어도 1개의 프로퍼티 값은 지정된 클래스의 인스턴스
	owl:hasValue	프로퍼티 값은 지정된 클래스의 인스턴스 또는 데이터 값
출현 횟수에 대한 제약	owl:minCardinality	지정된 값이 프로퍼티의 최소 출현 횟수
	owl:maxCardinality	지정된 값이 프로퍼티의 최대 출현 횟수
	owl:cardinality	지정된 값이 프로퍼티의 출현 횟수

라) 클래스 표현의 논리 조합

어떤 클래스를 정의할 때 다른 클래스의 논리 조합에 의해서 표현하는 방법이다. 이때의 논리 조합에는 교집합, 합집합, 여집합이 있으며 이들은 각각 ‘owl:intersectionOf, owl:unionOf, owl:complementOf’으로 표현한다. 교집합은 열거된 모든 조건들을 동시에 만족시켜야 한다. OWL Lite에서는 교집합만 사용할 수 있다.

74) Dean Allemang·James A. Hendler(저)/김성혁·박영택·추윤미(역), 『온톨로지 개발자를 위한 시맨틱 웹』, 사이텍미디어, 2008, 237쪽.

마) 클래스 간 관계의 기술

OWL에서 'rdfs:subClassOf'는 클래스 간의 계층 관계를 표현할 수 있다. 'owl:equivalentClass'는 주어 클래스와 목적어 클래스가 동일한 개체들의 집합을 외연으로 가지고 있음을 나타낸다. 이것은 양자가 같은 클래스라는 것을 의미하지는 않는다. 'owl:disjointWith'⁷⁵⁾는 주어 클래스와 목적어 클래스가 서로소 관계임을 표현한다.

바) OWL 프로퍼티 공리

(1) 등가 관계 프로퍼티(owl:equivalentProperty)

등가 프로퍼티는 하나의 프로퍼티와 다른 프로퍼티가 동일한 외연을 가지는 관계를 말한다. 이때 외연은 클래스의 인스턴스 집합을 말한다. 결국 프로퍼티의 인스턴스가 같은 관계를 말한다. 두 개의 속성이 같은 것이라는 사실은 'owl:sameAs'를 사용하여 기술할 수 있다. 그러나 'owl:sameAs'는 두 개의 개체에 대한 동일성을 나타내는 것으로 속성에 적용하기 위해서는 속성이 개체가 될 수 있는 OWL Full에서만 가능하다.

(2) 이행성 프로퍼티(owl:TransitiveProperty)⁷⁶⁾

속성 P가 이행적이고, 甲, 乙, 丙이라는 개체가 甲-P-乙과 乙-P-丙이라는 관계가 성립한다면 자동적으로 갑-P-병이라는 관계를 추론할 수 있다. 예를 들어 '甲이 乙의 조상이고, 乙이 丙의 조상이라면 甲이 丙의 조상'이라는 관계가 성립한다. 이런 것을 이행성이 있다고 한다. 그러나 '甲이 乙의 아버지이고, 乙은 丙의 아버지이면, 甲이 丙의 아버지'라는 관계는 성립하지 않는다. 아래 그림은 짚의李克用的 자손 중 일부를 그림으로 표현한 것이다.

오른쪽의 그림으로부터 왼쪽의 관계를 확인할 수 있다.

75) 비접합성, 서로소 관계라고 한다.

76) TransitiveProperty는 추이형 또는 이행성이라고 번역한다.

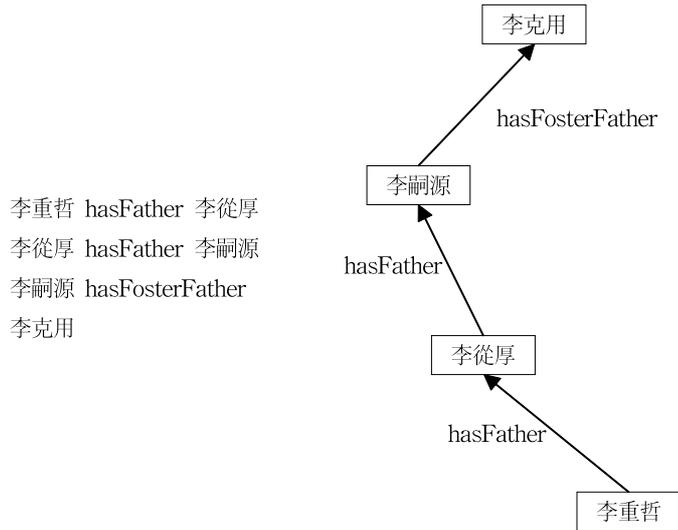


그림 14. 李克用 의 자손의 RDF 그래프 표현

이에 대해 이행 관계를 적용하기 위해서는 아래와 같이 rdfs:subPropertyOf와 rdf:type을 설정해 주어야 한다.

```

kno:hasFather rdfs:subPropertyOf kno:hasAncestor
kno:hasAncestor rdf:type owl:TransitiveProperty
  
```

이를 OWL로 표현하면 다음과 같다.

```

<owl:ObjectProperty rdf:about="&kno;hasFather">
  <rdfs:subPropertyOf rdf:resource="&kno;hasAncestor"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="&kno;hasAncestor">
  <rdf:type rdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>
  
```

위의 관계로부터 다음과 같은 관계를 추론할 수 있다.

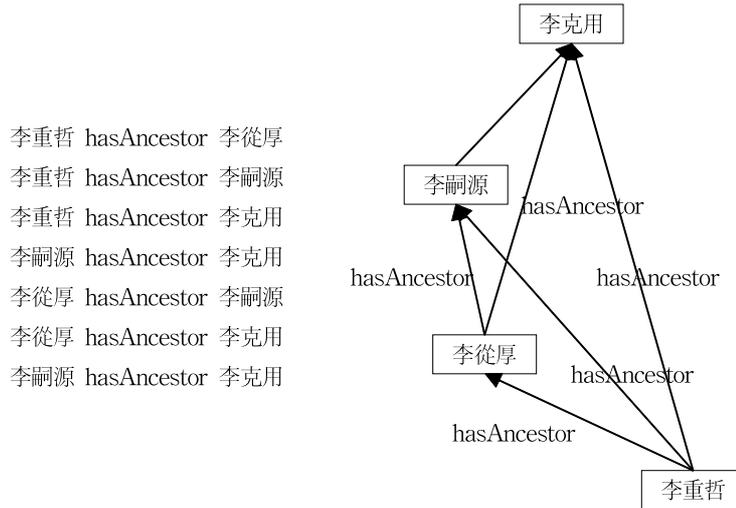


그림 15. 이행성 속성의 예77)

(3) 역 관계 프로퍼티(owl:inverseOf)⁷⁸⁾

역 관계는 어떤 프로퍼티가 다른 프로퍼티와 반대 관계에 있음을 표현하는 프로퍼티이다. 예를 들어 ‘唐 spatiallyInclude 京畿道(당나라는 경기도를 포함하고 있다.)’를 역으로 표현하면 ‘京畿道 spatiallyInside 唐(京畿道는 당나라 안에 있다.)’로 할 수 있다. 이런 관계를 역 관계라고 한다. 이런 예로 부분 관계를 표현하는 ‘hasPart’와 ‘isPartOf’, ‘hasMember’와 ‘isMemberOf’ 등 흔히 나타난다. 그림 아래와 같은 생부와 아들 간의 관계를 살펴 보자.

李存勖 hasBirthFather 李克用(李存勖은 生父로 李克用을 가진다)
 李克用 isBirthFatherOf 李存勖(李克用은 李存勖의 생부이다)
 李存勖 hasFather 李克用(李存勖은 아버지로 李克用을 가진다)
 李克用 hasSon 李存勖(李克用은 아들로 李存勖을 가진다)

hasBirthFather와 isBirthFatherOf는 역관계임에는 분명하다. hasFather와 hasSon의 관계를 역관계로 볼 수 있을까? 아래의 문장에서 관계를 보면 hasFather와 hasDoughter가 서로 역관계를 가지고 있는 것으로 보인다.

77) Dean Allemang·James A. Hendler(저)/김성혁 외(역), 앞의 책(2008), 144쪽을 참조로 하여 그림을 내용에 맞게 수정하였다.

78) inverseOf는 역 프로퍼티, 반대 관계 프로퍼티, 역의 관계 프로퍼티, 역 관계 프로퍼티 등으로 번역하였다.

瓊華長公主 hasFather 李嗣源(瓊華長公주는 아버지로 李嗣源을 가진다)

李嗣源 hasDoughter 瓊華長公主(李嗣源은 딸로 瓊華長公주를 가진다)

사실 이 문제는 논리적인 문제보다도 선언적인 문제로 접근해야 할 것으로 보인다. 즉 'hasFather와 hasSon' 혹은 'hasFather와 hasDoughter'를 역관계로 설정하면 거기에 맞춰 논리적인 추론을 해주는 것으로 보면 될 것이다. 주어와 목적어는 위치에 따라 능동과 피동으로 관계로 표현할 수도 있고, 용어 자체를 반대말을 사용하여 표현할 수도 있는 것이다.

(4) 대칭형 프로퍼티(owl:SymmetricProperty)

대칭형 프로퍼티(owl:SymmetricProperty)는 주어와 목적어를 바꿔 쓰더라도 같은 관계가 성립하는 성질을 가진다. 예를 들어 管仲과 鮑叔牙은 서로 친구인데 양 방향으로 모두 친구 관계가 성립하기 때문에 대칭형 프로퍼티로 표현할 수 있다. owl:inverseOf가 하나의 프로퍼티와 다른 프로퍼티를 연결하는 것이 비해, SymmetricProperty는 하나의 프로퍼티의 특징이며 하나의 클래스로서 OWL에서 표현된다.⁷⁹⁾

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFriendOf -->
<owl:ObjectProperty rdf:about="&kno;hasFriendOf">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:comment>친구 관계</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSocial"/>
</owl:ObjectProperty>
```

(5) 함수 프로퍼티(owl:FunctionalProperty)

함수 프로퍼티는 주어와 목적어의 관계 횟수를 제한할 수 있다. 이 프로퍼티를 적용하면 주어에 대응하는 목적어는 0개 또는 1개 출현할 수 있다. 예를들어 오대십국의 後唐 閔帝 李從厚의 생부는 李嗣源이며, 李嗣源은 후에 李暹으로 개명한다는 사실이 아래 그래프처럼 '李從厚의 생부는 李暹이다'와 '李從厚의 생부는 李嗣源이다'라는 두 개의 문장이 제시되었다고 하자. 이를 RDF 트리플로 표현하면 다음과 같다.

hasBirthFather rdf:type owl:FunctionalProperty

李從厚 hasBirthFather 李暹

李從厚 hasBirthFather 李嗣源

79) Dean Allemang·James A. Hendler(저)/김성혁 외(역), 앞의 책(2008), 140쪽.

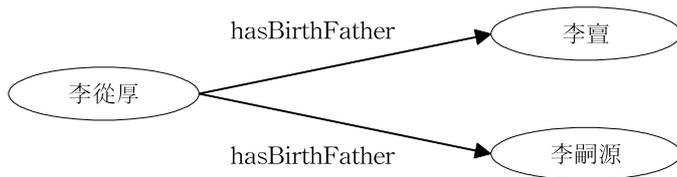


그림 16. 함수 프로퍼티의 사용 예

이를 통해서 '李亶 owl:sameAs 李嗣源'을 추론할 수 있다. 그런데 이런 함수 프로퍼티가 개체가 서로 같다는 것을 표현하는 일반적인 패턴은 아니다.

(6) 역함수 프로퍼티(owl:InverseFunctionalProperty)

역함수 프로퍼티는 특정 목적어에 대응하는 주어는 0개 또는 1개로 제한하는 프로퍼티이다. 이 프로퍼티는 관계형 데이터베이스로 말하면 중복을 허용하지 않는 PK(Primary Key)와 같다고 할 수 있다. 역함수 프로퍼티는 함수 프로퍼티와 밀접하게 관련되어 있다. 'hasBirthFather'가 함수 프로퍼티라면 그것의 역인 'isBirthFatherOf'는 역함수 프로퍼티가 된다. 주의해야 할 점은 역함수 프로퍼티는 'owl:ObjectProperty'의 하위 클래스라는 점이다. 이 말은 OWL DL과 OWL Lite에서는 Datatype Property를 역함수로 지정할 수 없다는 것을 의미한다.⁸⁰⁾ 함수와 역함수 프로퍼티 모두가 되는 것은 완전하게 1:1 관계를 의미한다. ID와 같은 것은 1:1이 되는 것이 바람직하기 때문에 이 공리를 조합하여 사용한다.

예를 들어보자. 오대십국의 後唐 明宗은 초명이 李嗣源이며 나중에 李亶으로 개명을 한다. 李嗣源의 생부는 李電이며, 후에 李嗣源은 晉王 李克用的 양자가 되었다. 그런데 만약 '李亶의 생부는 李電이다'와 '李嗣源의 생부는 李電이다'라는 것이 맞다면 이는 李亶과 李嗣源은 동일한 사람이 된다.

```

    hasBirthFather rdf:type owl:InverseFunctionalProperty
    李亶 hasBirthFather 李電
    李嗣源 hasBirthFather 李電
  
```

80) 앞의 그림 10을 참조. InverseFunctionalProperty는 rdf:type이며, owl:ObjectProperty의 하위 클래스이다.

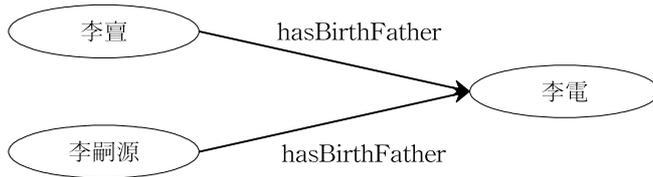


그림 17. 역함수 프로퍼티의 사용 예

위의 RDF 문장으로부터 ‘李賈 owl:sameAs 李嗣源’이라는 결론을 추론할 수 있다. 따라서 위에서 언급한 함수 프로퍼티와 역함수 프로퍼티를 활용하면 동명이인을 처리할 수 있다.

앞서 한 인물을 지칭하는 명칭은 姓名 이외에도 字, 號, 諡號, 廟號, 堂號, 法號, 法名, 初名, 初字, 本名, 別稱 등이 있음을 언급하였다. 이들이 동일한 인물을 지칭하고 있음을 표시하기 위해서 OWL의 기본 속성 중 속성의 출현 횟수에 대한 제약 속성인 ‘FunctionalProperty’를 사용한다. 그 이유는 일반적인 방법은 역함수 속성을 사용하는 것이나 인물의 여러 가지 명칭은 Datatype Property로 정의하였기 때문에 OWL DL에서는 역함수 속성을 사용할 수 없기 때문이다.

마. 시맨틱 웹과 LOD(Linked Open Data)

LOD(Linked Open Data)는 Semantic Web을 실현시키기 위한 방법 중 하나이다. 현재의 웹은 팀 버너스 리가 최초로 의도했던 웹과 달리 대부분 인간이 이해하는 문서를 하이퍼링크한 ‘문서의 웹(Web of Document)’이 주를 이루고 있다. 이러한 문서(Document)의 웹에서는 문서를 인간이 검색해 식별하고 이해해서 활용해야 하는데, 웹상의 문서를 모두 찾기도 힘들고, 정확성도 떨어져 활용이 제한적일 수밖에 없다. 이러한 웹의 한계를 극복하기 위해 팀 버너스 리는 웹상의 데이터와 데이터베이스를 서로 연결하자는 링크된 오픈 데이터(LOD, Linked Open Data) 프로젝트를 제안하면서 세계적으로 확대된 것이다.⁸¹⁾ LOD는 사회·사상적 측면에서 오픈된 데이터를 서로 연결(Link)해 가치를 만들어 나가는 것이며, 기술적 측면에서 기계 가독성이 있는 구조화된 데이터를 생성해서 의미있게 연결하는 것을 의미한다.⁸²⁾

81) Berners-Lee, T. The next web. Retrieved from,

https://www.ted.com/talks/tim_berniers_lee_on_the_next_web, 2009)

82) 한국정보화진흥원, 「LOD 기반의 데이터 관리 패러다임 전환 전략」, IT & Future Strategy(1호)

LOD는 웹의 한계를 극복하기 위해 문서(Document) 단위가 아닌 기계가 이해할 수 있는 데이터(Data)를 기본 단위로 하여 하이퍼링크로 연결하는 ‘데이터의 웹(Web of Data)’을 추구한다. 여기서 문서는 HTML 형식을 말하며, 데이터는 RDF 형식을 말한다. 시맨틱 웹은 정보와 정보를 연결하기 위해서 기존의 Hypertext, HTML 대신 RDF로 표현되는 새로운 정보 표현 방식과 상호 연결(interlinking)이라는 방법을 사용하여 연결한다.

RDF는 데이터(자원)를 주어-술어-목적어로 표현하고 있기 때문에 이를 RDF 트리플이라 하며 지식을 담고 있는 최소 구성 요소라고 할 수 있다. RDF는 지식 세계를 이루고 있는 온갖 사물과 개념들을 자원(resource)으로 보고 이러한 자원의 식별자로 URI 참조를 사용하여 자원이 가지고 있는 속성이나 자원과 자원 간의 관계를 기술하는 데이터 모델⁸³⁾이다.

표 9. ‘문서의 웹’과 ‘데이터의 웹’ 차이점⁸⁴⁾

문서의 웹(Web of Document)	LOD에 의한 데이터의 웹(Web of Data)
<ul style="list-style-type: none"> ▪ 문서 파일(HTML) + 링크 ▪ 각 문서의 소재를 URI로 표기 ▪ 인간이 읽는 것을 상징(데이터 활용의 한계) ▪ 문서를 키워드로 검색, 필요 정보 취득 	<ul style="list-style-type: none"> ▪ 데이터(RDF) + 링크 ▪ 각 데이터의 소재를 URI로 표기 ▪ 기계가 읽는 것을 상징(데이터 연결, 처리, 활용이 용이) ▪ 데이터를 SQL 같은 조건식으로 검색, 필요 정보 취득

LOD는 ‘사람만 이해하는 웹’에서 ‘기계가 이해하는(의미처리) 웹’으로 전환하기 위한 기법으로, 데이터를 누구나 이용할 수 있는 형식으로 개방해 링크시켜 나가는 구조를 말한다. LOD는 데이터를 각각 연계하고(Linked Data), 이를 오픈된 RDF 형식으로 웹상에 개방·연계해서(Outlink) 상호 공유함으로써(Linked Open Data) 새로운 가치 창출할 수 있다. 팀 버너스 리는 2009년 2월에 개최된 TED 컨퍼런스에서 ‘Design Issue: Linked Data’ 강연을 통해 LOD의 구축에 필요한 원칙을 다음과 같은 네 가지를 제시하였다.⁸⁵⁾ 첫 번째는 개체 식

2014. 4, 6쪽.

83) 노상규·박진수 저, 앞의 책(2009), 122쪽.

84) 한국정보화진흥원, 「LOD 기반의 데이터 관리 패러다임 전환 전략」, IT & Future Strategy(1호)

2014. 4, 6쪽.

85) 팀 버너스 리, 「The next web」, TED 강연(온라인 참조 :

별을 위한 URI를 사용하는 것이며, 두 번째는 HTTP 프로토콜을 사용해서 데이터 정보에 접근할 수 있게 하는 것이다. HTTP 프로토콜을 이용하여 URI로 명명된 사물에 접근할 수 있게 하며 이때 HTTP URI는 역참조(dereferenceable)가 가능해야 한다. 세 번째는 URI를 참조할 때 RDF, SPARQL(Simple Protocol and RDF Query Language) 등의 표준 기술을 사용해야 하며, 네 번째는 더 많은 개체를 연결할 수 있도록 개체에 타 URIs 링크를 포함해야 한다는 것이다.⁸⁶⁾

2. 선행 연구 및 사례 분석

가. 선행 연구

역사에서 하나의 사건은 주제, 시간, 대상, 장소, 행위 등과 같은 개체 간의 작용과 반작용의 과정으로 어떤 상황(situation) 또는 상태를 변화시키며, 상황은 객체들과 그들의 관계로서 이루어 진다. 또한 사건 간에는 복잡한 의미 관계를 형성하고 있으며, 사건이 일어나는 원인이나 특정 사건에 의해 발생하는 결과의 표현을 위해서는 역사적 지식이 사건을 중심으로 표현되어야 한다. 그런데 이러한 사건은 RDFS, OWL 등과 같은 다양한 온톨로지 언어로 표현할 수 있다. 특히 OWL은 지식의 표현을 위해 개념(class), 개체(individual), 속성(property), 개념 간의 관계(relationship), 개념의 범위를 지정하기 위한 제약 사항(restriction), 그리고 string, integer 등과 같은 데이터(data) 등의 온톨로지 요소를 정의하고 있어 한층 더 복잡한 구조를 표현할 수 있다.⁸⁷⁾ 그러나 이러한 언어를 이용하여 사건을 단순히 Subject-Predicate-Object 형태의 개념간의 관계로만 표현하기에는 역사적 사실은 너무 다양하고 복잡하다.

이인근 등은 이를 개선하려는 방법으로 시간 개념이 포함된 역사적 지식을 사건 기반으로 구축하고 활용할 수 있는 온톨로지 모형인 Y-HisOnto를 제안하였다. 제안한 온톨로지 모형은 이진 관계를 조합하여 다진 관계(n-ary relationship)를 표현하는 사건 온톨로지(event ontology)를 이용하였다. 예를 들어 “전봉준을 중심으로 신분제도철폐를 위해 조선에서 일어난 민중 봉기인 동학운동은 청일전쟁의 원인이 되었다”라는 역사적 사건 지식을 아

http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html, 검색일자 2017년 7월 15일)

86) Tim Berners-Lee, Linked Data, <http://www.w3.org/DesignIssues/LinkedData.html>

87) 이인근·정재은·황도삼, 『Y-HisOnto: Q&A 시스템에서의 활용을 위한 역사 온톨로지 모형』, 『한국정보과학회언어공학연구회 2013년도 제25회 한글 및 한국어 정보처리 학술대회』, 2013, 156쪽.

래 그림과 같은 그래프로 표현할 수 있다. 이들 간의 관계는 일차 논리(First-order logic)를 이용하여 P(S, O) 형태로 정리하면 ‘주체(동학운동, 민중), 주체(동학운동, 전봉준), 장소(동학운동, 조선), 인과(동학운동, 청일전쟁), 시작시간(동학운동, 1893), 종료시간(동학운동, 1895), 목적(동학운동, 신분제도철폐)’로 나타낼 수 있다.⁸⁸⁾

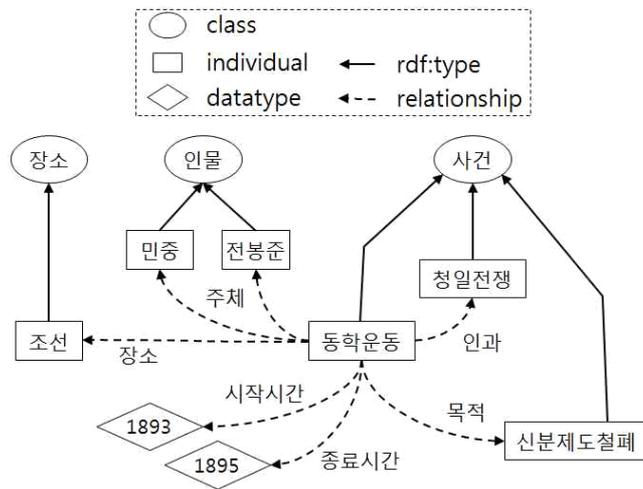


그림 18. 역사적 사건 지식의 온톨로지 표현⁸⁹⁾

이 논문은 이진 관계를 조합하여 복잡한 사건을 표현하려고 시도하였으며, 사건에 시간 개념을 포함한 사건 지식 및 그들 사이의 인과 관계를 형성하기 위한 방법을 제안하고, 그 활용 가능성을 확인하였다는 것에 의미가 있다고 할 수 있다. 그런데 이러한 방법은 ‘동학운동’처럼 하나의 완결된 사건을 중심으로 발생 시간이나 인과 관계 등을 표현하는 것은 가능하지만, 아래와 같은 하나의 구체적인 행위를 표현하기는 적절치 않다. 아래 문장은 ‘시간’과 ‘공간’, ‘가해자’, ‘피해자’, ‘발생 장소’, ‘구체적 행위’ 등이 나타나 있다.

11월 丙子日(13일)에 楊那 등이 들어와 조회할 적에 甲士 수십 명이 廣政殿으로부터 뛰어나와서 楊那·史弘肇·王章을 동쪽 행랑채 아래에서 죽였다.

88) 이인근 외, 앞의 논문(2013), 157쪽.

89) 이인근 외, 위의 논문(2013), 157쪽.

표 10. 사건과 관련한 부가 정보

시간	11월 13일
구체적 시점	楊邠 등이 들어와 조회할 적
공간	廣政殿 동쪽 행랑채 아래
가해자	甲士 수십 명
대상	楊邠·史弘肇·王章
행위	죽였다

위의 문장은 아주 간단한 문장이나 RDF 트리플로 표현하는 것은 쉽지 않다는 문제점이 있다.

최주원은 영어 문장을 온톨로지 형태로 자동 변환하기 위해 기존에 어노테이션 된 자원과 입력 문장의 유사도를 이용하여 술어-논항 구조 어노테이션을 하는 방법을 제시하였다. 이때 술어 논항 구조는 특정 동사와 그 동사와 대응하는 주어, 목적어, 수식어를 묶어서 지칭한다.⁹⁰⁾ 아래 박스의 (1)에서 broke는 ‘부수는 사람’과 ‘부서지는 물건’을 논항으로 가질 수 있다. 이들을 ARG0, ARG1로 표현한다. (2)에서 The window는 주어의 위치에서 부서지는 것에 대응하므로 ARG1으로 표시한다. 술어는 웹에서 만들어지고 있는 수많은 자연언어 문서들을 사용한다.

(1) [John]ARG0 brokePRED [the window]ARG1
(2) [The window]ARG1 brokePRED
PRED : 술어, ARG : 논항

그리고 온톨로지 매핑과 관련하여 워드넷⁹¹⁾이나 SUMO⁹²⁾같은 대규모 지식베이스와 연결시켜 유사도를 측정하는 방안을 제시하였다. 워드넷은 많은 어휘를 포함하기 때문에 문장에서 나오는 개념들을 대부분 매칭할 수 있고, 또한 이는 SUMO와 연결되기 때문에 더욱 많은 추론의 여지가 있다고 하였다.

박현규는 시간 정보를 시점(point)과 기간(interval)로 분류한 후, 기간은 시작 시점과 종료 시점으로 이루어진다고 하였다. 또한 복잡한 문장을 RDF의 구체화(reification)를 사용하여 표현하였다. 예를 들어 “나는 아인슈타인이 1933년부터 1955년까지 미국에서 살았다는 사실을 wiki에서 발견하였다.”라는 문장은 아래 그림과 같이 그래프 모델로 표현하였다.⁹³⁾

90) 최주원, 『문서 기반 온톨로지 생성을 위한 문장 분해 및 유사도 측정 시스템』, KAIST 석사학위 논문, 2009, 13쪽.

91) WordNet, Princeton University, <https://wordnet.princeton.edu>

92) Suggested Upper Merged Ontology, <http://www.adampease.org/OP/>

93) 박현규, 「대용량 온톨로지를 위한 시간 정보 표현 및 추론」, 숭실대학교 대학원 석사학위논문,

구체화 방법을 사용하기 위해 Statement1(아인슈타인이 미국에서 살았다)과 Statement2(나는 발견하였다 Statement1)를 기술하고, 각각은 subject, predicate, object로 주어, 술어, 목적어를 기술하였다. 아인슈타인이 미국에 살았던 기간을 표현하기 위해 occurs 프로퍼티로 interval과 연결하였다. 이때 발생한 기간은 시점과 종점을 Point로 연결하고 Point와 구체적인 시간은 time로 연결하였다.

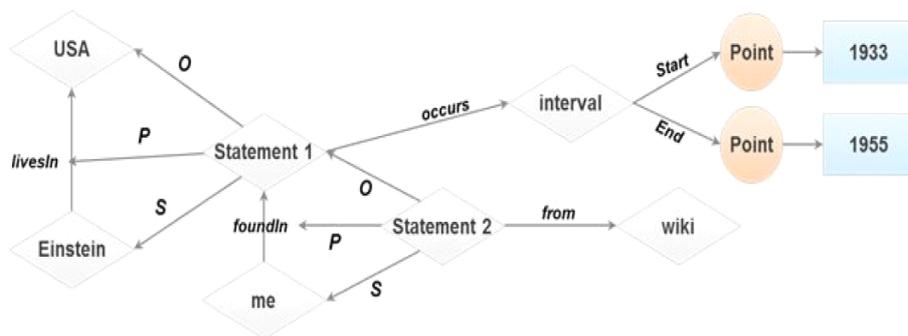


그림 19. 기간을 표현한 복문의 예⁹⁴⁾

이 모델은 구체화라는 방법을 통해서 복잡한 복문을 표현할 수 있게 되었으나 시간 이외의 다양한 정보를 추가하기에는 너무 복잡해 진다는 문제가 있다.

Radinsky 등은 「Learning Causality for News Events Prediction」에서⁹⁵⁾ 150년 간의 뉴스 보도를 수집한 후 이벤트를 ‘Action, Actor, Object, Instrument, Location, Time’로 표현하는 방식으로 3억 개 이상의 사실 노드가 10억 개 이상의 엣지로 연결된 시맨틱 구조의 인과 관계 그래프를 생성하였다. 그리고 이를 이용한 예측 알고리즘 만들어서 현재 사건을 고려하여 가능한 영향을 예측할 수 있는 예측자 g를 출력하는 학습 알고리즘을 만들었다. 예를 들어 2004년 10월에 발생한 “미군이 폭발물로 이라크에 있는 창고를 파괴했다”라는 사건은 아래와 같은 구성 요소와 RDF 그래프로 표현할 수 있다.

- Destroy(Action), 미국 육군(Actor), 웨어 하우스(Object), 폭발물(Instrument), 이라크 (Location), 2004년 10월(Time)

2014, 22쪽.

94) 박현규, 위의 논문(2014), 24쪽.

95) Radinsky, Kira, Sagie Davidovich, Shaul Markovitch, 「Learning to predict from textual data」, 『Journal of Artificial Intelligence Research』, 45, 2012, p915.

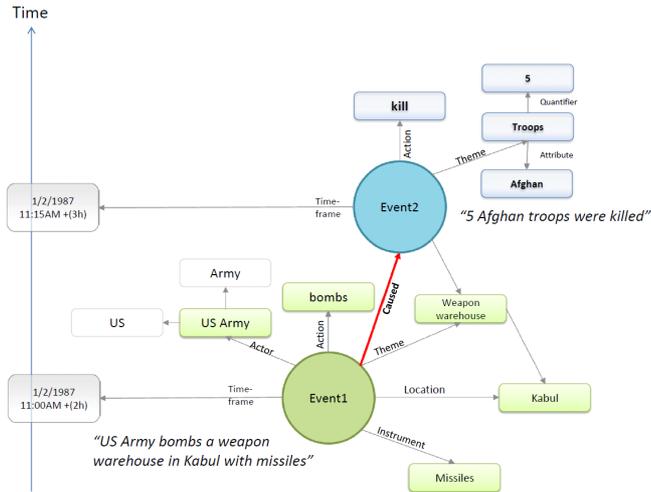


그림 20. 인과 관계 그래프⁹⁶⁾

위의 구조는 이벤트를 중심에 두고 이벤트와 나머지 인스턴스의 관계를 Action, Theme 등의 프로퍼티로 표현하였으며, 구체적인 행위는 Action 인스턴스로 표현하였다. 이러한 방법을 활용함으로써 좀 더 복잡한 자연어 문장을 어느 정도 표현할 수 있게 되었다.

나. 사례 분석

1) 국내 LOD 발행 현황

온톨로지 설계와 어휘의 재활용을 위해서는 현재까지 LOD로 발행된 사례를 참조해야 한다. ⁹⁷⁾ 우선 온톨로지의 구조와 사용된 어휘 등을 참조해야 하며, 실제 데이터 셋을 다운받거나 SPARQL을 이용하여 직접 조회하여 어떤 내용인지 확인해야 한다.

현재까지 구축된 LOD 현황에 대해서는 한국정보화진흥원에서 간행한 「2014 링크드 오픈 데이터 국내 구축 사례집」을 참조할 수 있다. 이 자료는 수행기관, 서비스 유형, 접근 경로, 서비스의 개요, 구축 절차, 소프트웨어 구성도, 활용어휘 등이 일목요연하게 정리되어 있어 실제로 LOD 시스템을 구축할 때 많은 도움이 된다. 이를 토대로 현재까지 서비스 되고 있는 LOD 현황은 다음과 같다.

96) Radinsky, Kira, Sagie Davidovich, Shaul Markovitch, 「Learning to predict from textual data」, 『*Journal of Artificial Intelligence Research*』, 45, 2012, p915.

97) 한국정보화진흥원, 「2014 링크드 오픈 데이터 국내 구축 사례집」, 2014.

표 11. 국내 LOD 발행 현황

LOD	URL	제공
Bible Ontology	http://bibleontology.com	주식회사 리스트
KDATA	http://kdata.kr	주식회사 리스트
K-water LOD	http://opendata.kwater.or.kr/lod/main	한국수자원공사
KISTI LOD	http://lod.ndsl.kr/home/	한국과학기술 정보연구원
moviedata LOD	http://moviedata.kr	주식회사 리스트
RISS LOD	http://data.riss.kr/main.do	한국교육학술정보원
국립공원 LOD	http://lod.knps.or.kr/main/	국립공원관리공단
국립중앙도서관 LOD	http://lod.nl.go.kr	국립중앙도서관
산업재산권 LOD	http://lod.kipo.kr/	특허청
문화융합 LOD	http://lod.culture.go.kr/sparqlexample.do	한국문화정보원
생물정보 LOD	http://lod.naris.go.kr/service/wildlife/intro.do	국립수목원, 국립중앙과학관
서울시 열린 데이터 광장 LOD	http://lod.seoul.go.kr/home/intro/openLod.jsp	서울시
주소데이터 LOD	http://jusodata.kr/main/index.jsp	주식회사 리스트
한국사 LOD	http://lod.koreanhistory.or.kr	국사편찬위원회
한국관광공사 LOD	http://data.visitkorea.or.kr/linked_open_data	한국관광공사

2) 한국사 LOD

한국사 LOD는 한국사와 관련하여 유관 기관에서 보유하고 있는 데이터를 융합하여 한국사에 관한 다양한 서비스나 어플리케이션 개발에 활용할 수 있도록 LOD 형태로 제공하는 서비스이다.⁹⁸⁾ 여기에는 국사편찬위원회의 인물 자료와 시소러스, 한국학중앙연구원의 한국민족문화대백과사전, 한국역대인물정보, 문화재청의 유물·유적데이터 등이 포함되어 있다. 원 소스는 XML, RDBMS 등의 형식으로 존재하는 것을 RDF triple로 변환하여 LOD로 서비스하고 있다.⁹⁹⁾ 특이한 점은 각 키워드별로 전문가가 핵심적 원문 사료를 인용하여 집필한 해설을 함께 제공하고 있다는 것이다.

한국사 LOD의 클래스 정의는 크게 ‘Event, Information, Organization, Person, Relic, TopicDBInfo’ 등 6개의 범주로 구성되어 있다.¹⁰⁰⁾ Person과 Organization은 foaf(Friend of

98) 한국정보화진흥원, 「2014 링크드 오픈 데이터 국내 구축 사례집」, 2014, 30쪽.

99) 한국사 LOD, 국사편찬위원회, <http://lod.koreanhistory.or.kr/lodIntro.do>

100) 온톨로지를 분석하기 위해서는 온톨로지 모델과 최종 RDF 데이터 셋을 확인해야 한다. 보통 온톨로지 모델을 근거로 하여 RDF 트리플을 생성하는 과정에서 모델과 실데이터가 일치하지 않는 경우가 발생하여 이 단계에서 모델의 수정이 빈번하게 발생하기도 한다. 한국사 LOD도 웹

a Friend)¹⁰¹⁾ 클래스를, 주제명 표현을 위해 SKOS¹⁰²⁾ 어휘를 활용하였으며, 나머지는 역사 자료에 적합하도록 새롭게 구성한 것이다. 이 중 본 연구와 관련이 깊은 사건을 중심으로 살펴보면 한국사 LOD에서 사건과 관련된 클래스는 이벤트(Event)이며, 이 이벤트는 다른 클래스와 ‘hasEtcEvent, hasEvent, RelatedPerson, hasHistoricalMaterialInfo, hasPostEvent, hasPreEvent, hasSuperEvent’ 등의 Object Property로 관계를 표현하였으며, ‘culturalCategory, encyKoreaCategory, eventOccurPlace, materialKind, materialName, relatedLaw, relatedPolicy, relatedSystem’를 Datatype Property로 연결이 되어 있다.

특징을 살펴보면 ‘hasEventObjectProperty, hasOrgObjectProperty, hasPersonObjectProperty, hasRelicObjectProperty’는 Object Property로 설계하였으며, Datatype Property에도 동일한 내용을 ‘hasEventDatatypeProperty, hasOrgDatatypeProperty, hasPersonDatatypeProperty, hasRelicDatatypeProperty’으로 명명하여 속성을 Datatype Property로 연결하였다. 이것은 이들 속성으로 연결되는 목적어가 클래스 형식과 리터럴 방식 중 어느 하나만으로는 표현할 수 없기 때문인 것으로 보인다. 사건과 관련이 있는 오브젝트 프로퍼티는 ‘hasRelatedOrganization(관련 조직), hasRelatedPerson(관련 인물) 등이 있으며, 시간과 관련하여 hasStartAge(시작일), hasendDate(종료일)이, 공간과 관련하여 eventOccurPlace(사건 발생 장소)가, 분류 정보로 eventkind(사건의 종류) 등이 있으며 기타 relatedPerson(관련인물)¹⁰³⁾ 등의 정보로 데이터와 연결을 시키고 있다. 전체적으로 보면 Property는 클래스 간의 관계를 표현하기 위해 ObjectProperty 53개와 클래스와 데이터 간의 관계를 표현하기 위한 DatatypeProperty 89개로 구성되어 있다.¹⁰⁴⁾

사이트에서 제공하는 온톨로지 모델과 실제 데이터 셋 사이에 큰 차이가 있어 본 논문은 최종 데이터를 기준으로 어휘를 다시 정리하였다.

101) Dan Brickley, Libby Miller, “FOAF Vocabulary Specification 0.99”, 2014 :

<http://xmlns.com/foaf/spec>

102) W3C, SKOS Simple Knowledge Organization System Namespace Document - HTML Variant, 2009, <http://www.w3.org/2004/02/skos/core#>

103) 2017년 3월 현재 국사편찬위원회의 한국사 LOD 홈페이지에서 LOD 모델로 제시한 내용과 실제 데이터 상에는 약간의 차이가 있다. 예를 들어 RDF 모델에서 제공하는 정보 중에

‘hasEventPlace’가 ‘eventOccurPlace’로 변경이 되었으며, ‘beginDate’가 ‘hasStartDate’로,

‘ComingEndDate’가 추가된 것 등등 변경이 발생하였다.

104) 국사편찬위원회에서 제공하는 모델에 따르면 이러한 수치가 나오나 추가된 것이 있기 때문에 정확한 수치는 파악하기 어렵다.

표 12. 한국사 LOD 클래스 정의

클래스		내용
Evnet		사건
Information		관련정보
	OrganizationInformation	조직정보
	ArtworkInformation	저작물정보
	BadgeInformation	서훈정보
	CareerInformation	관직정보
	CeremonyInformation	행사정보
	ContentsInformation	콘텐츠정보
	DocumentaryInformation	생산기록물정보
	EducationInformation	교육정보
	EventInformation	사건정보
	GovernmentComingInformation	출사정보
	HistoricalMaterialInformation	사료정보
	HistoryInformation	연혁정보
	HomepageInformation	홈페이지정보
	OrganizationMemberInformation	조직구성원정보
	PersonInformation	인물정보
	PrisonInformation	수형정보
	RelatedInformation	Interlinking 관련정보
	RelicInformation	유물정보
	SourceInformation	출처정보
	ThoughtInformation	사상정보
Organization		조직단체
Person		인물
Relic		유물, 유적
TopicDBInfo		주제별 DB

표 13. ObjectProperty

속성		내용
hasContentsInfo		콘텐츠정보
hasEventObjectProperty		
	hasEtcEvent	기타연관사건
	hasEvent	사건정보의사건
	hasEventRelatedPerson	사건과연관된인물
	hasCultureEtcPerson	문화 - 기타
	hasCultureOppositionPerson	문화 - 반대
	hasCultureProsecutionPerson	문화 - 추진
	hasDiplomacyEtcPerson	외교 - 기타
	hasDiplomacyOpponentPerson	외교 - 상대방
	hasDiplomacyOursPerson	외교 - 우리측

		hasDiplomacyRelCountryPerson	외교 - 관련국
		hasEconomyEtcPerson	경제 - 기타
		hasEconomyOppositioPerson	경제 - 반대
		hasEconomyProsecutionPerson	경제 - 추진
		hasLawEtcPerson	법, 정책, 제도 - 기타
		hasLawOppositionPerson	법, 정책, 제도 - 반대
		hasLawProsecutionPerson	법, 정책, 제도 - 추진
		hasPoliticsEtcPerson	정치 - 기타
		hasPoliticsLeadingPerson	정치 - 주도
		hasPoliticsSubjectPerson	정치 - 대상
		hasSocietyEtcPerson	사회 - 기타
		hasSocietyLeadingPerson	사회 - 주도
		hasSocietySubjectPerson	사회 - 대상
		hasSocietySupressionPerson	사회 - 진압
		hasWarDefenceAggressorPerson	전쟁 - 방어측 동조국 인물
		hasWarDefencePerson	전쟁 - 방어
		hasWarEtcPerson	전쟁 - 기타
		hasWarOffenceAggressorPerson	전쟁 - 공격측 동조국 인물
		hasWarOffencePerson	전쟁 - 공격
	hasHistoricalMaterialInfo		관련사료
	hasPostEvent		후행사건
	hasPreEvent		선행사건
	hasSuperEvent		상위사건
hasHistoryInfo			관련연혁정보
hasOrgObjectProperty			
	hasAssociatedOrganization		관련조직
	hasOrganizationMemberInfo		조직구성원
	hasProducedDocumentaryInfo		생산기록물정보
	hasSuperOrganization		상위조직
hasPersonObjectProperty			
	hasAdmirer		사숙
	hasArtworkInfo		관련저작물정보
	hasBadgeInfo		관련서훈정보
	hasCareerInfo		관련관직정보
	hasCareerRelPerson		관직관계자
	hasCompanionship		교유
	hasDisciple		제자
	hasEducationInfo		관련교육정보
	hasFamily		가족
		hasBrother	형제
		hasChild	자녀
		hasConcubine	첩
		hasConcubineFather	첩부

		hasFather	부
		hasFatherInLaw	치부
		hasGrandFather	조부
		hasGrandMother	조모
		hasGrandSon	손자
		hasGreatGrandFather	증조부
		hasMaternalGrandFather	외조부
		hasMother	모
		hasOtherRelation	외, 다수
		hasRealFather	생부
		hasRealMother	생모
		hasRoyalConcubine	후궁
		hasSister	누이
		hasSonInLaw	사위
		hasWife	배우자
	hasGovComingInfo		관련출사정보
	hasGovComingRelPerson		출사관계자
	hasPrisonInfo		관련수형정보
	hasTeacher		스승
hasRelatedEvent			관련사건
hasRelatedEventInfo			관련사건정보
hasRelatedOrganization			관련조직
hasRelatedOrganizationInfo			관련조직정보
hasRelatedPerson			관련인물
hasRelatedPersonInfo			관련인물정보
hasRelatedRelic			관련유물
hasRelatedRelicInfo			관련유물정보
hasRelicObjectProperty			
	hasCeremonyInfo		관련행사정보
	hasHighRelic		상위유물
	hasLowRelic		하위유물
	hasThoughtInfo		관련사상정보
hasSourceInfo			관련출처정보
homepageInfo			관련홈페이지정보

표 14. Data type Property

속성		내용
contentsName		콘텐츠명
culturalInfo		관련유물문화재정보
culturalPlace		관련유물문화재지역
currentAddress		현주소

currentLocation		현재지명
detailExplanation		상세설명
eventName		사건명
explanation		설명
hasAge		시대
hasDate		년월일
hasDynasty		왕대
hasEndAge		종료시대
hasEndDate		종료년월일
hasEndDynasty		종료왕조
hasEventDatatypeProperty		
	culturalCategory	문화재청분류
	encyKoreaCategory	민족문화대백과사전분류
	eventOccurPlace	사건발생장소
	materialKind	사료유형
	materialName	사료명
	relatedLaw	관련법
	relatedPolicy	관련정책
	relatedSystem	관련제도
hasExternalLink		설명URL
hasOrgDatatypeProperty		
	achievement	구성원업적
	definition	정의및설명
	division	국립/공립/사립구분
	documentaryAuthor	생산기록물저자
	documentaryNameChi	생산기록물명칭한자
	documentaryNameEng	생산기록물명칭영어
	documentaryNameKor	생산기록물명칭한글
	documentaryPublish	생산기록물게재처
	documentaryType	생산기록물유형
	mainAgent	설치주체
	organizationType	조직단체유형
	purpose	목적및강령
	thenLocation	당시지명
hasPersonDatatypeProperty		
	ancestry	골품
	artworkName	저작물명
	artworkPublish	게재처
	artworkPublisher	출판사
	artworkType	저작물 유형
	badgeName	서훈명
	birthDate	출생 년월일
	birthPlace	출생지
	buName	부명
	careerName	관직명
	ComingEndDate	재위 종료일
	deathDate	사망 년월일
	deathPlace	사망지

	educationKind		교육유형
	educationName		교육명칭
	educationOversea		유학
	familyPlace		본관
	gender		성별
	govComingKind		출사종류
	govComingRank		출사등수
	livingPlace		거주지
	name		
		aliasName	이칭
		bonghoName	봉호
		changedName	개명
		childhoodName	아명
		choName	초명
		entryName	표제어명
		jaName	자
		myohoName	묘호
		nickName	호, 아호
		penName	필명
		posthumousName	시호
		realName	본명
		realNameChi	본명(한자)
		relegionName	종교명
	nationalPlace		인물국가
	occurAge		주요활동시기
	occurDynasty		주요활동왕대
	occurPlace		활동지
	personType		인물유형
	prisonName		형벌명
	prisonPlace		형벌지역
	prisonReason		형벌이유
	registerPlace		본적
	reignStartDate		재위시작일
	religionName		종교명
	tombPlace		묘, 장지
hasRelicDatatypeProperty			
	ceremonyName		행사명
	culturalAppointDate		문화재지정일
	culturalCode		문화재번호
	culturalType		문화재구분
	historyName		연혁명
	materialInfo		재질정보
	originalLocation		원소재지
	ownedName		소장이름
	ownedUrl		소장URL
	sizeInfo		크기정보
	thoughtName		사상명
	webSite		웹사이트

hasRole		역할
hasStartAge		시작시대
hasStartDate		시작년월일
hasStartDynasty		시작왕조
homepageName		홈페이지명
homepageUrl		홈페이지URL
koreanHistoryCategory		한국역사정보통합시스템시소러스분류
organizationName		조직명
organizationRole		조직역할
relatedCountry		관련국가
relicCulturalInfo		유물문화재정보
relicName		유물명
relicPlace		유물지역
sourceName		출처명
title		
	afterNameChi	이후명칭한자
	afterNameEng	이후명칭영어
	afterNameKor	이후명칭한글
	afterNameOtherFL	이후명칭기타어
	anotherNameChi	이칭한자
	anotherNameEng	이칭영어
	anotherNameJpn	이칭일본어
	anotherNameKor	이칭한글
	beforeNameChi	이전명칭 한자
	beforeNameEng	이전명칭 영어
	beforeNameKor	이전명칭 한글
	beforeNameOtherFL	이전명칭기타어
	nameCalledByChi	타국에서부르는 명칭 한자
	nameCalledByEng	타국에서부르는 명칭 영어
	nameCalledByJan	타국에서부르는 명칭 일본어
	nameCalledByKor	타국에서부르는 명칭 한글
	titleChi	대표명 한자
	titleEng	대표명 영어
	titleKor	대표명 한글
	titleOtherFL	대표명 기타 외국어
topicCategory		주제 분류
topicDB		
	thesaurusData	주제별 DB 시소러스
	topicDBContents	주제별 DB 콘텐츠
	topicDBHeadCopy	주제별 DB 헤드카피
	topicDBHistoricalMaterial	주제별 DB 사료정보
	topicDBIndex	주제별 DB 내용
	topicDBMultimedia	주제별 DB 멀티미디어
	topicDBRelatedInfo	주제별DB 관련 정보
	topicDBType	주제별 DB 타입

다음은 본 연구에서 중요한 대상으로 다루고 있는 사건이 한국사 LOD에서는 어떤 키워

드로 표현되었는지 확인하기 위해 국사편찬위원회의 사건으로 분류된 RDF 트리플을 확인해 보았다. 한국사 LOD는 주제별 핵심적 원문 사료를 인용하여 전문 인력을 투입하여 수작업으로 RDF 트리플을 만들었으므로 그 정확성과 신뢰도가 보장된다고 할 수 있다.¹⁰⁵⁾ 그러므로 사건으로 분류된 RDF에서 사용된 키워드를 분석하는 것은 의미가 있다고 할 수 있다. 사건으로 분류된 RDF는 대략 2,000개 정도가 된다.¹⁰⁶⁾ 이들을 대상으로 사건을 구성하는 어휘 중 핵심어를 추출해 본 결과 ‘戰鬪, 海戰, 戰爭, 侵入, 征伐, 蜂起’ 등의 군사적인 행위와 ‘冊封, 論爭, 獄死, 卽位, 彈劾, 廢位’ 등 정치적 사건이 가장 큰 비중을 차지하고 있었다. 이러한 키워드는 한국사 뿐만 아니라 중국사에서도 대부분 유사하게 적용될 것으로 보이며, 역사적 사건에 대한 일반적인 인식으로 생각된다.

표 15. 한국사 LOD의 사건에 사용된 용어의 빈도

키워드	빈도	키워드	빈도	키워드	빈도	키워드	빈도	키워드	빈도	키워드	빈도
戰鬪	129	死亡	18	築城	12	告變	8	謀議	6	試圖	5
事件	96	戰爭	17	蜂起	11	論難	8	民亂	6	押送	5
設置	82	卽位	17	遷都	11	復舊	8	復位	6	入朝	5
冊封	55	廢止	17	建議	10	擁立	8	收復	6	整備	5
亂	48	改定	16	歸國	10	築造	8	要求	6	除去	5
制定	28	獄	16	誣告	10	開設	7	入手	6	條約	5
要請	24	建立	15	頒布	10	反亂	7	沮止	6	主管	5
運動	24	征伐	15	任命	10	弑害	7	政變	6	處刑	5
海戰	24	派遣	15	創建	10	遠征	7	處罰	6	逐出	5
論議	23	施行	14	滅亡	9	執權	7	被殺	6	討伐	5
論爭	23	彈劾	14	謀反	9	編纂	7	開通	5	播遷	5
獄死	22	廢位	13	伏誅	9	改革	6	攻擊	5	歸附	4
實施	21	殺害	12	宿衛	9	禁止	6	大捷	5	反對	4
侵入	20	逆謀	12	讒訴	9	企圖	6	迫害	5	變	4
結婚	19	創刊	12	建國	8	來投	6	肅清	5	批判	4

다음으로 한국사 LOD에서 사용한 URI 참조를 살펴보자. 한국사 LOD에서 사용한 URI 체계는 ‘인명_순종, 단체_법상종, 사건_정유제란’처럼 클래스 명과 인스턴스를 결합한 형태로 식별자를 부여하였다. 유물정보와 같이 이름만으로 식별하기 어려운 경우에는 일련번호를 사용하였다.

105) 김지영·박선아·이선희, 「과학인물 LOD 구축에 관한 연구」, 『한국도서관·정보학회지』, 제45권, 제4호, 2014, 434쪽.

106) <http://lod.koreanhistory.or.kr/lodIntro.do>에서 2017년 3월에 제공하는 데이터를 기준으로 분석하였다.

```

<RDF:Description RDFabout="http://lod.koreanhistory.or.kr/인명_이순신">
  <hlod:hasRelatedRelicInfo RDFresource="http://lod.koreanhistory.or.kr/유물정보_59215" />
</RDF:Description>
<RDF:Description RDF:about="http://lod.koreanhistory.or.kr/단체_법상중">
  <hlod:hasRelatedRelicInfo RDF:resource="http://lod.koreanhistory.or.kr/유물정보_55298"/>
</RDF:Description>
<RDF:Description RDF:about="http://lod.koreanhistory.or.kr/사건_경유재판">
  <hlod:hasRelatedRelicInfo RDF:resource="http://lod.koreanhistory.or.kr/유물정보_21122"/>
</RDF:Description>

```

그리고 클래스명과 이름으로 중복이 발생하는 경우에는 동일한 어휘를 구분하기 위해 ‘인명_순종, 인명_순종2’와 같이 일련번호를 부착하여 구분해 주었다.

```

<RDF:Description RDFabout="http://lod.koreanhistory.or.kr/인명_순종">
  <hlod:tdRelatedInfo RDFresource="http://lod.koreanhistory.or.kr/TD_HUMAN_선종" />
</RDF:Description>
<RDF:Description RDFabout="http://lod.koreanhistory.or.kr/인명_순종2">
  <hlod:tdRelatedInfo RDFresource="http://lod.koreanhistory.or.kr/TD_HUMAN_순정효황후" />
</RDF:Description>

```

한국사 LOD의 인터링킹 현황을 살펴보면 한국교육학술정보원 RISS LOD¹⁰⁷⁾, Korea DBpedia¹⁰⁸⁾, 국립중앙도서관의 국가서지 LOD, wikidata 등과 ‘owl:sameAs’을 이용하여 약 2천 6백여 건이 인터링킹으로 연결되었으며, 외부 링킹은 daum, 한국근현대인물자료¹⁰⁹⁾, 한국고전번역원의 고전번역서¹¹⁰⁾, 한국문집총간¹¹¹⁾, 한국민족문화대백과사전¹¹²⁾ 등의 사이트와 ‘hlod:hasExternalLink’(외부 URL)을 이용하여 약 4만 3천 여건의 자원을 연결하였다.

그 형식을 살펴보면 다음과 같다.

107) RISS linked open data, 한국교육학술정보원, <http://data.riss.kr>

108) DBpedia Korea, <http://ko.dbpedia.org/page>

109) 한국근현대인물자료, 국사편찬위원회, <http://db.history.go.kr/item/level.do?itemId=im>

110) 고전번역서, 한국고전번역원,

<http://db.itkc.or.kr/dir/item?itemId=BT#/dir/list?itemId=BT&gubun=book>

111) 한국문집총간, 한국고전번역원,

<http://db.itkc.or.kr/dir/item?itemId=MO#/dir/list?itemId=MO&gubun=book>

112) 한국민족문화대백과사전, 한국학중앙연구원, <http://encykorea.aks.ac.kr>

```

<RDF:Description RDF:about="http://lod.koreanhistory.or.kr/인명_가실왕">
<owl:sameAs RDF:resource="http://ko.dbpedia.org/page/%EA%B0%80%EC%8B%A4%EC%99%95"/>
<owl:sameAs RDF:resource="http://www.wikidata.org/entity/Q10855025"/>
</RDF:Description>
<RDF:Description RDF:about="http://lod.koreanhistory.or.kr/인명_강공훤">
<hlo:hasExternalLink RDF:resource="http://encykorea.aks.ac.kr/Contents/Index?contents_id=E0000996"/>
<hlo:hasExternalLink RDF:resource="http://encykorea.aks.ac.kr/Contents/Index?contents_id=E0036419"/>
</RDF:Description>

```

3) YAGO

많은 어플리케이션들이 온톨로지를 활용한다. 그런데 대부분의 어플리케이션은 오직 하나의 지식 온톨로지만을 사용할 수밖에 없었다. 이로 인해 기존의 지식을 융합한 하나의 거대한 온톨로지의 필요성을 공감하게 되었다. YAGO(Yet Another Great Ontology)¹¹³⁾ 온톨로지는 이러한 필요성을 충족시키기 위해 등장한 온톨로지로서, 위키피디아(Wikipedia) 카테고리 계층의 말단 카테고리 와 워드넷(WordNet)¹¹⁴⁾의 계층 정보를 매핑하여 구축한 지식 베이스이다¹¹⁵⁾. YAGO는 사람(person), 장소(location)와 같은 개체(entities) 정보를 170만 개 이상 포함하고 있으며¹¹⁶⁾, 두 개체가 하나의 관계(relation)를 맺고 있는 트리플 단위의 사실(facts) 정보는 1,500만 개 이상 포함하고 있다. YAGO 모델에서는 사실이 발견된 출처 정보를 함께 저장하기 위해 사실 식별자(fact identifier)를 도입하고, 이를 통해 구체화 그래프(reification graph) 형태로 온톨로지를 표현하였다. 예를 들어 “項羽의 이름은 ‘籍’이며, 기원전 232년에 태어나 기원전 202년에 사망했다”는 사실을 위키피디아에서 발견했다”는 내용을 구체화 그래프로 표현하면 다음과 같이 표현할 수 있는데, 아래에서 #1, #2, #3은 사실 식별자(fact identifier)에 해당하고, 사실 식별자는 실제 사실 앞에 표시 된다. ‘項羽,

113) YAGO, Max Planck Institute for Informatics,

<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago>

114) WordNet RDF, Princeton University, <http://wordnet-rdf.princeton.edu>

115) 강신재, 「데이터의 웹을 위한 상호연결된 대규모 온톨로지 네트워크 구축」,

『한국산업정보학회논문지』, 제15권, 제1호, 2010, 18쪽.

116) Download the Entire YAGO, YAGO,

<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads>

Wikipedia' 등은 인스턴스에, 'time, source'은 프로퍼티에 해당한다. 이러한 방법을 통해서 RDF의 기본 규칙인 2항 관계로 표현하기 힘들었던 N항(n-ary) 관계를 표현할 수 있게 되었다.

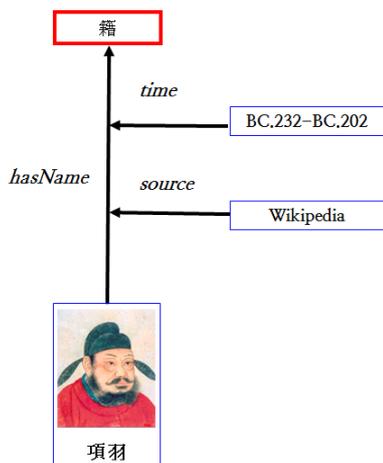


그림 21. 項羽를 이용한 N항 관계의 표현

#1 (項羽, hasname, 籍)
#2 (#1, time, BCE.232~BCE.202)
#3 (#1, source, Wikipedia)

현재는 YAGO 온톨로지의 확장인 YAGO2가 등장하였는데, 이 온톨로지는 YAGO에 GeoNames를 추가로 융합하여 기존에 불충분했던 시·공간상의 정보를 보완하여 제공하는데 초점을 맞추었다.¹¹⁷⁾

4) EuroWordnet

유로워드넷(EuroWordnet)은 미국의 심리언어학자 G.A. Miler가 개발한 워드넷(Wordnet) 1.5에¹¹⁸⁾ 기초하여 만들어진 유럽 8개국 언어의 다국어 어휘 의미 데이터베이스이다. 기본

117) 강민서·김재성·김선동·이재길, 「한글 DBpedia 온톨로지 스키마 구축」, 『한국정보과학회언어공학연구회 2014년도 제26회 한글 및 한국어 정보처리 학술대회』, 2014, 140쪽.
 118) 프린스턴대학교 인지과학 연구소의 George A. MilLer에 의해 고안되고 구성된 영어

적으로 각 단어의 의미가 다른 단어 의미들과 관계에 의해서 기술되는 관계적 의미 데이터베이스이다. 워드넷은 영어를 대상으로 구성된 어휘 의미망이지만 유로워드넷은 네덜란드어, 이탈리아어, 독일어, 불어, 에스토니아어, 에스파냐어, 영어, 체코어 등 다국어어를 기반으로 하고 있다는 점이 다르다.¹¹⁹⁾

유로워드넷의 온톨로지는 최상위항을 제외하고 63개의 상위 개념으로 구성되어 있다. Lyons¹²⁰⁾에 따르면 첫 번째 층위에서 3유형의 실체를 구별할 수 있다. 먼저 1차 실체는 감각에 의해 지각할 수 있고, 3차원의 공간에서 어떤 시점에 위치하고 있는 구체적인 실체를 말한다. 여기에 속하는 것들은 항상 구체 명사로 표현된다. 2차 실체는 어떤 상태적 상황(속성, 관계)이나 동적 상황을 말하며, 이는 독립적인 물리적 사물처럼 잡거나 듣거나 보거나 느낄 수 없다. 2차 실체는 시간 속에서 위치할 수 있고, 존재한다기보다 일어나거나 발생한다. 3차 실체는 시간과 공간과는 독립적으로 존재하는 명제이다. 3차 실체는 실재하는 것이 아니라 참이거나 거짓일 수 있다. 또한 단정되거나 부인되거나 잊혀질 수 있다. 1차 실체와 2차 실체는 계층구조에 따라 다시 분류가 된다.

1차 실체는 구체물을 개념화하거나 분류하는데 기원(Origin), 형상(Form), 구성(Composition), 기능(Function)이라는 네 가지 방법에 따라 구분된다. 기원은 실체가 생겨난 방법을 말하며, 형상은 무정형 실체, 또는 고정된 형태를 가진 물체를 말한다. 구성은 자신을 포함한 전체의 그룹으로서 또는 그러한 전체의 일부로서 Part와 Group으로 다시 구분되었다. 기능은 전형적인 행위 또는 실체와 연관된 행동을 말한다.

2차 실체는 상태적 상황 또는 동태적 상황을 의미하는 명사와 동사, 형용사와 부사를 이용하여 지시된다. 모든 2차 실체는 상황 유형(SituationType)과 상황 성분(SituationComponent)이라는 두 개의 서로 다른 분류화 스키마를 이용하여 분류된다. 상황 유형은 사건 구조를 말하며, 하나의 상황은 시간에 걸쳐 있는 개념적 단위로서 특성화된다. 상황 성분은 하나의 상황을 특징짓는 가장 두드러진 의미적 성분을 말하며, 용도(Usage), 시간(Time), 사회성(Social), 수량(Quantity), 목적(Purpose), 소유(Possession), 물리적(Physical), 양태(Modal), 정신(Mental), 방식(Manner), 처소(Location), 경험(Experience), 존재(Existence), 외적 상태(Condition), 의사소통, 원인(Cause), 작인성(Agentive), 자극(Stimulating), 현상(Phenomenal) 등으로 구분되어 있다.¹²¹⁾

워드넷이다. 이 워드넷은 한 언어의 단어 집합에 대해 같은 의미를 갖는 어휘들의 모음인 동의어 집합(synonym set)을 명사, 동사, 형용사, 부사의 각 품사별로 별도로 정의하고, 신셋 간의 의미 관계를 표현하고 있다. 이숙의, 『한국어 동사 온톨로지 구축 연구』, 역락, 2013, 24~25쪽.

119) 박종호, 『한국어 동사 의미망 구축 방법론』, 청운, 2012, 25쪽.

120) Lyons, J. 『Semantics』, Cambridge University Press, 1977.

121) Vossen Piek(편)/한정환·최경봉·도원영·이봉원·이동혁·이현희·김혜영(역), 『유로워드넷』,

☞ 16. The EuroWordNet Top-Ontology¹²²⁾

Top ⁰	
1stOrderEntity ¹	2ndOrderEntity ⁰
Origin ⁰	SituationType ⁹
Natural ²¹	Dynamic ¹³⁴
Living ³⁰	BoundedEvent ¹⁸³
Plant ¹⁸	UnboundedEvent ⁴⁸
Human ¹⁰⁶	Static ²⁸
Creature ²	Property ⁶¹
Animal ²³	Relation ³⁸
Artifact ¹⁴⁴	SituationComponent ⁰
Form ⁰	Cause ⁶⁷
Substance ³²	Agentive ¹⁷⁰
Solid ⁶³	Phenomenal ¹⁷
Liquid ¹³	Stimulating ²⁵
Gas ¹	Communication ⁵⁰
Object ¹⁶²	Condition ⁶²
Composition ⁰	Existence ²⁷
Part ⁸⁶	Experience ⁴³
Group ⁶³	Location ⁷⁶
Function ⁵⁵	Manner ²¹
Vehicle ⁸	Mental ⁹⁰
Representation ¹²	Modal ¹⁰
MoneyRepresentation ¹⁰	Physical ¹⁴⁰
LanguageRepresentation ³⁴	Possession ²³
ImageRepresentation ⁹	Purpose ¹³⁷
Software ⁴	Quantity ³⁹
Place ⁴⁵	Social ¹⁰²
Occupation ²³	Time ²⁴
Instrument ¹⁸	Usage ⁸

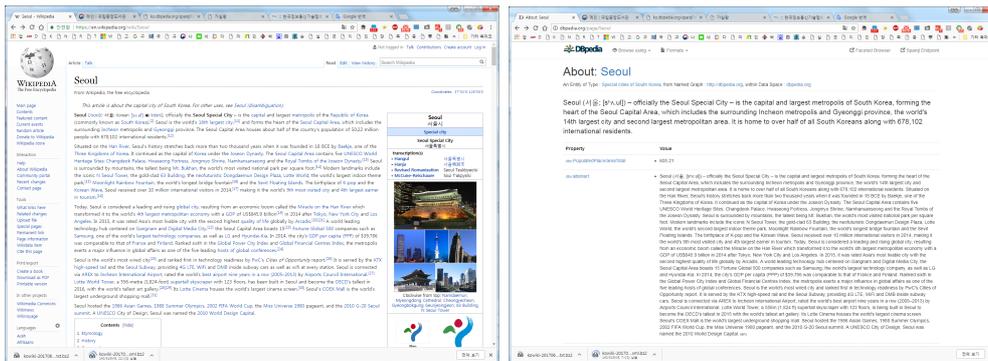
한국문화사, 2004, 116~124쪽.

122) Vossen, Piek, 「Version 3 Final July 1, 2002 Piek Vossen (ed.) University of Amsterdam」, 2002, 61쪽.

Garment ³ Furniture ⁶ Covering ⁸ Container ¹² Comestible ³² Building ¹³	
3rdOrderEntity ³³	

5) DBpedia

DBpedia¹²³⁾는 베를린 자유 대학과 라이프치히 대학의 공동 연구로 2007년도에 시작되었다. 이 프로젝트는 Wikipedia로부터 구조 데이터를 자동으로 추출해 RDF로 변환하고, SPARQL Endpoint를 통해 링크드 데이터의 퍼블리싱, 저장, 접근 및 관리 기능을 제공하고 있다. 위키피디아(Wikipedia)는 소수의 전문가에 의해 생산되는 지식보다 다수의 협업과 경쟁을 통해 토론과 합의를 거친 지식이 더 우월하다는 신념을 바탕으로 누구나 자유롭게 참고하고 집필할 수 있는 인터넷 무료 백과사전이다.¹²⁴⁾ 이 두 개의 데이터는 아래 그림에서 볼 수 있듯이 서로 큰 차이가 없어 보이나 ‘구조화된 데이터’라는 측면에서 보면 상당히 다르다. 디비피디아는 이름에서도 알 수 있듯이 디비 형태의 구조화된 데이터이다. 정확히 말하면 RDF Triple 형태로 되어 있다.



<http://wikipedia.org/wiki/Seoul>

<http://dbpedia.org/resource/Seoul>

그림 22. Dbpeia와 WikiPedia 비교

DBpedia는 링크드 데이터의 가장 중심에서 허브 역할을 하고 있고, 많은 사이트와 연결

123) DBpedia, <http://wiki.dbpedia.org>

124) 정승화, 「인터넷 백과사전 위키피디아와 지식권력의 재구성」, 『人文科學』, 106권, 2016, 87쪽.

관계를 유지하고 있으며, 링크드 데이터를 실현하는 데 매우 중요한 역할을 수행하고 있다.¹²⁵⁾ DBpedia가 중요한 이유는 위키피디아의 데이터를 정형화된 지식으로 제공한다는 것 뿐만 아니라, 웹상의 다른 데이터셋을 링크로 통합하고 있기 때문이다.¹²⁶⁾

현재 영어, 독일어, 일본어 등 다양한 언어로 DBpedia를 구축하였으며 이것을 활용한 오픈소스 어플리케이션들이 쏟아져 나오고 있다. 그 이유는 DBpedia는 접근이 쉽고, 비교적 정확하며, 무엇보다도 시간이 지날 수록 정보가 축적되는 장점이 있기 때문이다.¹²⁷⁾

아래 그림은 DBpedia를 중심으로 빨간색으로 다른 LOD 데이터 간 연결된 현황을 보여주고 있다.

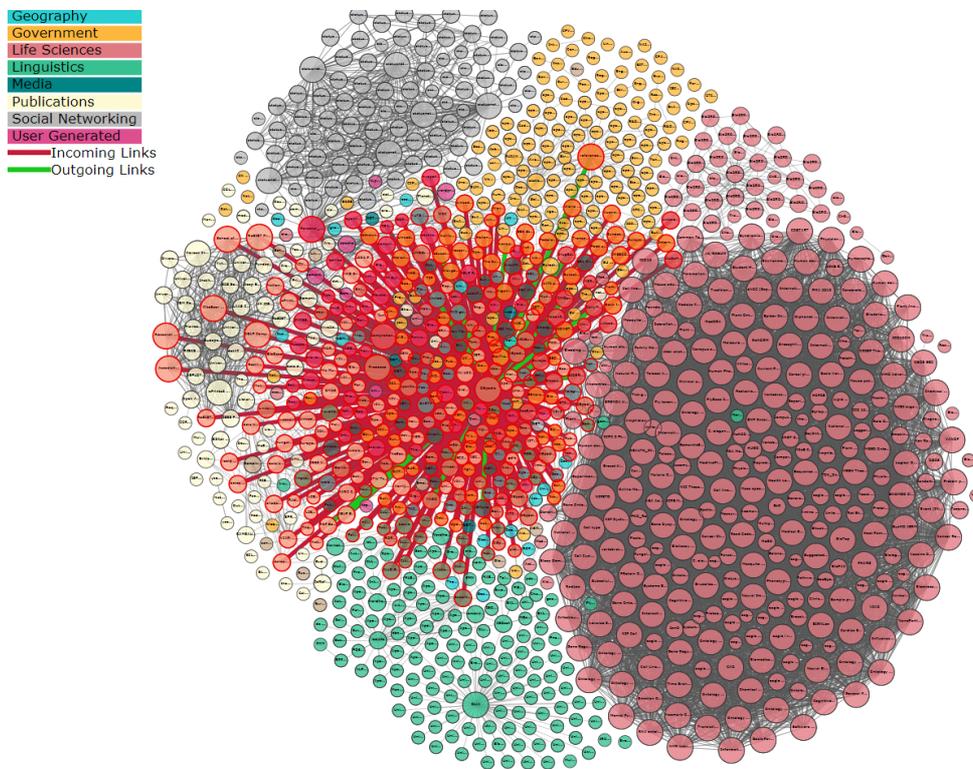


그림 23. 링크드 오픈 데이터 연결 현황

125) 박원주·남궁현·조기성·류원, 「링크드 데이터 기술 기반의 공공 오픈 데이터 기술 동향」, 『정보통신산업진흥원주간기술동향』, 2013, 4~5쪽.

126) 이현정, 「링크드 데이터 활용을 위한 멀티소스 검색시스템 구축에 관한 연구」, 중앙대학교 대학원 석사학위논문, 2015, 57쪽.

127) 임경태·함영균·Rezk Martin·박정열·윤용운·최기선, 「데이터 웹을 위한 한국어 DBpedia리소스 접근방법 연구」, 『한국정보과학회 2012 가을 학술발표논문집』, 제39권, 제2호(B), 2012, 112쪽.

DBpedia 온톨로지 구조를 살피기 위해서는 OWL 형태로 제공하는 DBpedia Ontology¹²⁸⁾를 참조해야 한다. 영어 위키피디아를 토대로 한 영문 DBpedia 온톨로지는 스키마(owl)와 인스턴스가 모두 잘 구축이 되어 있다. 그리고 영어 DBpedia의 각 Class에 한글 레이블이 달려 있다. 하지만 한글 레이블이 없는 영어 DBpedia Class가 절반이 넘는다.¹²⁹⁾ DBpedia에서 설계한 온톨로지는 인물과 시간의 경우 역사서를 온톨로지로 표현하는 데도 유용하므로 사람의 출생과 사망, 그리고 시간 등은 이 온톨로지 어휘를 참조하였다.

3. 도구의 활용

온톨로지는 시맨틱 웹의 핵심이다. 온톨로지를 쉽게 만들고 관리하기 위해서는 반드시 적절한 도구를 활용해야 한다. 이러한 도구 없이 직접 온톨로지 구문을 프로그래밍 한다면 온톨로지는 정말로 어려운 노동이 될 것이다. 다행스럽게도 근래에는 수많은 온톨로지 도구가 개발되어 유·무료로 제공되고 있다. 처음에는 단순한 편집 기능 위주로 개발되었으나 점차 진화하여 다양한 기능을 지원하고 있어, 많은 부분 인간의 노력을 줄여 주고 있다. 그런데 이제는 우리가 접할 수 있는 도구가 너무 많아서 어떤 도구를 사용해야 할지 선택하는 것이 중요한 문제로 대두되었다. 이 장에서는 본 논문에서 온톨로지 설계를 위해 사용한 도구를 소개한다. 먼저 RDF 구문을 생성하기 위한 텍스트 베이스의 에디터는 Altova Semanticworks를 주로 사용하였으며, RDF 구문의 파싱¹³⁰⁾은 W3C에서 제공하는 Validation Service과 OWL Validator를 활용하였다. 온톨로지 모델링은 Pretege4, Pretege5, TopBraid Composer를 활용하였으며, 온톨로지의 그래프 표현은 WebVOWL과 gruff¹³¹⁾를 주로 활용하였다.

가. Protégé

Protégé는¹³²⁾ 미국 스탠포드 의과대학에서 지식 기반 구조를 온톨로지로 작성하기 위해 개발한 온톨로지 설계 도구이다. Protégé는 오픈 소스로 무료로 공개되어 있어 누구든지 무료로 이용할 수 있다. 1987년 최초로 개발된 이후 지속적으로 프로그램을 개선해 왔으며 현

128) DBpedia Ontology는

‘<http://wiki.dbpedia.org/downloads-2016-04#h26493-2#dbpedia-ontology>’에서 다운받을 수 있다.

129) 강민서·김재성·김선동·이재길, 「한글 DBpedia 온톨로지 스키마 구축」,

『한국정보과학회언어공학연구회 2014년도 제26회 한글 및 한국어 정보처리 학술대회』, 2014,

139쪽.

130) W3C, Validation Service, <https://www.w3.org/RDF/Validator/rdfval>

131) Franz Inc, <http://franz.com/agraph/gruff>

132) Protégé, <http://Protégé.stanford.edu>

재 Protégé 5.2.0¹³³⁾까지 공개되었다. Protégé는 로컬 PC에 설치해서 사용할 수 있는 프로그램과 웹에서 사용할 수 있는 WebProtégé도 제공하고 있다.¹³⁴⁾ Protégé는 필요에 따라 여러 가지 플러그 인을 설치하여 기능을 확장할 수 있다. 예를 들어 인스턴스 간의 관계를 보기 위해서는 ProtégéVOWL을 설치하면 된다.¹³⁵⁾ Protégé에 대한 자세한 정보는 'http://protege.stanford.edu'에서 확인할 수 있다.

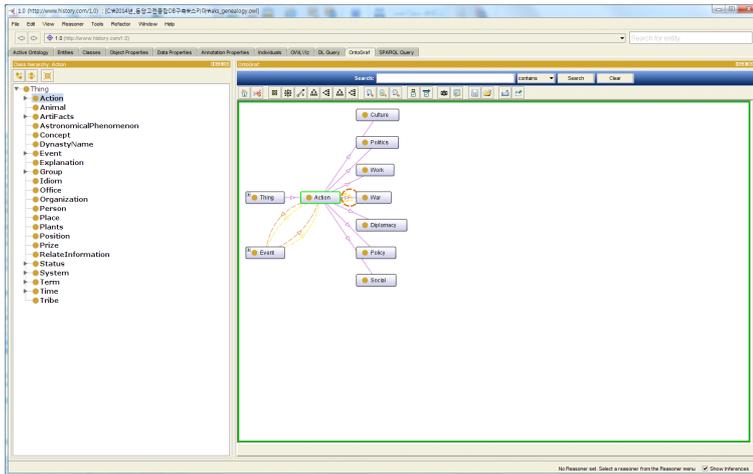


그림 24. Protégé 실행 화면

나. Altova SemanticWorks¹³⁶⁾

Altova SemanticWorks는 XMLspy로 유명한 Altova에서 개발한 시맨틱 웹 도구이다. Altova SemanticWorks는 앞에서 설명한 Protégé와 달리 상업용이며, 홈페이지에서 30일 평가판을 다운받아 기능을 테스트해 볼 수 있다. 대부분의 RDF 에디터가 텍스트 모드를 지원하지 않는 데 비해 Altova SemanticWorks는 텍스트 모드로 직접 RDF 구문을 수정할 수 있다.

Altova SemanticWorks는 RDF, RDF Schema, OWL Lite, OWL DL, OWL Full 등의 언어를 선택할 수 있다. syntax check와 semantic check 등 두 가지 파싱 기능을 제공한다. Altova SemanticWorks를 이용하면 온톨로지 설계를 시각적으로 할 수 있다. 화면 상단에

133) 2017년 3월 23일 현재 Protégé 5.2.0까지 공개되었다.

134) WebProtégé 2.6.0, <https://github.com/Protégéproject/webProtégé/releases/tag/v2.6.0>

135) VOWL Plugin for Protégé, <http://vowl.visualdataweb.org/Protégévowl.html>

136) Altova XMLSpy, <https://www.altova.com>

Classes, Properties, Instances, AllDifferent, Ontologies 탭에서 어느 하나를 선택하여 마우스 오른쪽 버튼을 눌러 클래스나 프러퍼티를 추가하면서 시각적으로 온톨로지를 설계할 수 있다. 화면 하단의 탭에는 text와 RDF/OWL을 선택하여 그래프 모드와 텍스트 모드를 선택할 수 있다.

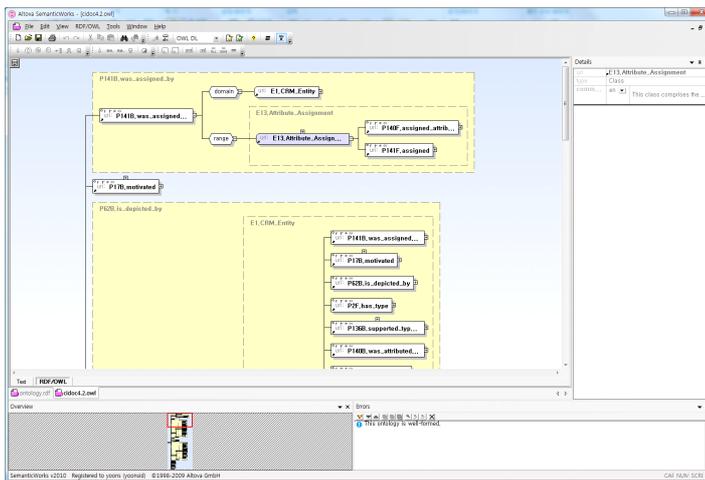


그림 25. Altova SemanticWorks 실행 화면

다. Validation Parse

Validation Parse¹³⁷⁾는 W3C에서 제공하는 RDF 파싱 프로그램이다. 웹을 통해서 RDF 문서의 오류를 점검할 수 있다. 파싱한 결과에서 오류가 발생한 라인을 표시해 주며 파싱한 결과를 트리플 형태, 트리플과 그래프 형태, 그래프 형태 중 하나를 선택해서 볼 수 있다.

137) W3C, Validation Service, <https://www.w3.org/RDF/Validator>

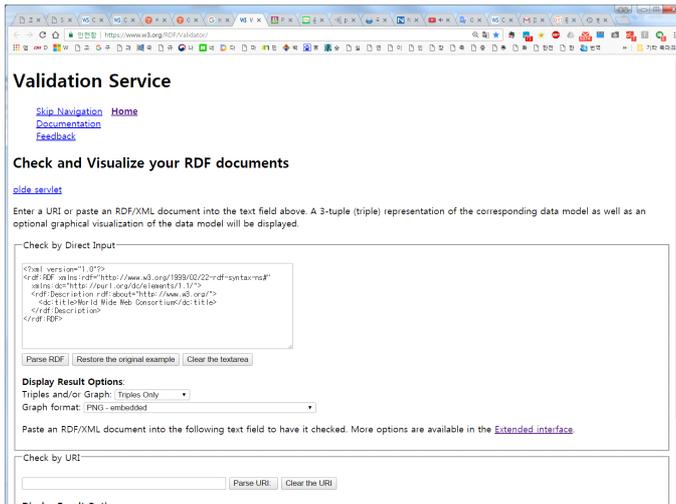


그림 26. W3C에서 제공하는 Validation Parse

OWL Validator¹³⁸⁾는 RDF Validator와 유사하게, 텍스트 박스에 OWL 구문을 붙여 넣거나 URI 주소를 지정하고 검증하면 OWL 문법 규칙을 준수했는지 체크해 준다. OWL의 종류를 선택할 수 있다.

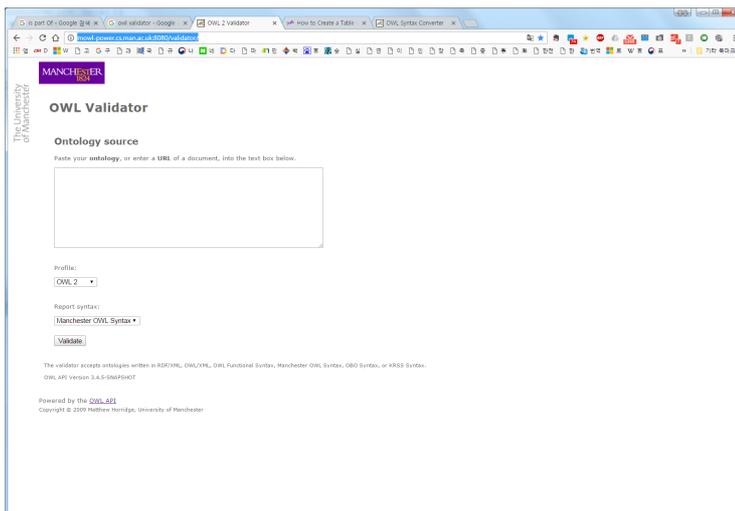


그림 27. OWL Validator

138) OWL Manchester, OWL Validator, <http://mowl-power.cs.man.ac.uk:8080/validator/>

그 밖에도 ‘OWL Syntax Converter’에서는 OWL 문서를 ‘RDF/XML’로 변환해 주는 서비스를 제공한다.¹³⁹⁾

라. TBC

TBC(TopBraid Composer)¹⁴⁰⁾는 미국 최초의 시맨틱 웹 회사인 탑쿼드란트(TopQuadrant)사에 의해 개발된 온톨로지 설계 도구이다. OWL 모델 파일을 읽어 직접 생성, 수정, 삭제를 할 수 있다. 유료이며 30일 동안 사용할 수 있는 트라이얼 버전을 제공한다.

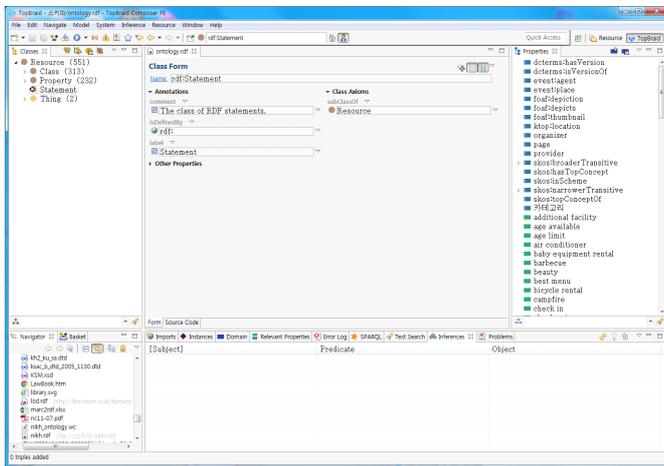


그림 28. TopBraid Composer 실행 화면

마. WebVOWL

WebVOWL(Web-based Visualization of Ontologies)¹⁴¹⁾은 온톨로지 모델을 시각적으로 확인할 수 있는 기능을 제공한다. 웹 사이트에 접속하여 ‘Run WebVOWL’을 클릭하면 바로 사용할 수 있다. 화면 하단에 있는 ontology 버튼을 누르고 OWL 파일을 선택한 후 upload 버튼을 눌러 파일을 업로드하면 클래스 간의 관계를 시각적으로 확인할 수 있다. 클래스 간 간격과 색상을 조정하고 필터하는 기능이 있어 대략적인 설계 모델을 확인할 때 유용하게 활용할 수 있다. Pretege의 플러그인으로 사용할 수 있는 ProtégéVOWL¹⁴²⁾을 다운받아 활

139) OWL Syntax Converter, <http://mowl-power.cs.man.ac.uk:8080/converter/>

140) TopQuadrant, <http://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition>

141) visualdataweb, <http://visualdataweb.de/webvowl>

용할 수도 있다.

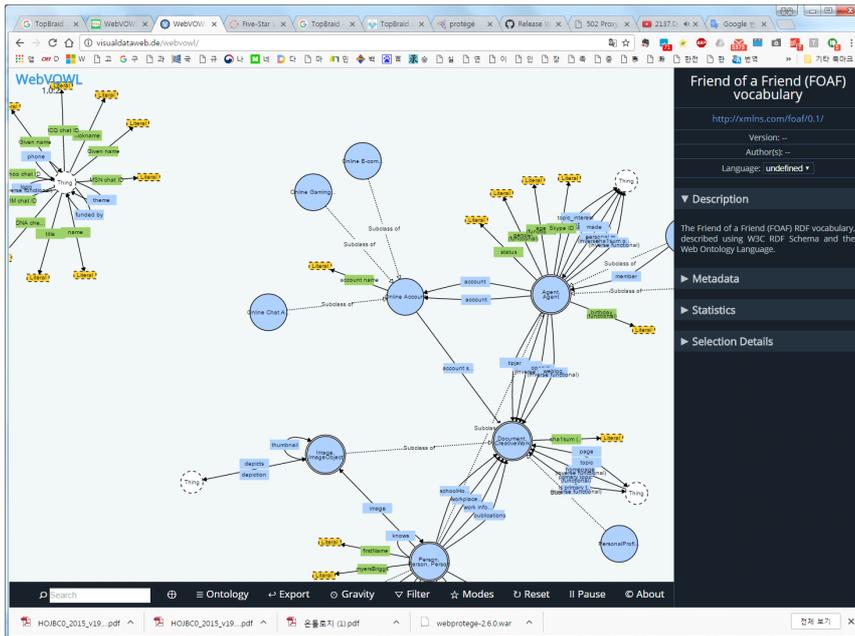


그림 29. WebVOWL 실행 화면

142) VOWL, Visual Notation for OWL Ontologies, <http://vowl.visualdataweb.org/protegevowl.html>

III. 역사 자료 연계 및 통합 방안

이 장에서는 본 연구의 대상 자료인 『通鑑節要』에 대해서 먼저 알아보고, 일반적인 온톨로지 설계 절차를 알아본다. 설계 절차는 먼저 개체명에 대해서 파악을 하고 이를 근거로 클래스와 프로퍼티를 도출한다. 또한 온톨로지 어휘의 재활용을 위해 기존에 발행된 LOD 어휘와 온톨로지 라이브러리 사이트를 참조하여 작성된 모델을 수정 및 보완한다.

1. 『通鑑節要』과 『資治通鑑』

이 장에서는 본 논문의 주제인 역사 자료의 연계 및 통합 방법을 모색하기 위해 대상 자료로 선정한 『通鑑節要』¹⁴³⁾와 『通鑑節要』의 저본이 된 『資治通鑑』을 살펴 본다. 『通鑑節要』는 北宋 시대 江贇(?~?)¹⁴⁴⁾가 司馬光(1019-1086)의 『資治通鑑』¹⁴⁵⁾을 節錄하여 편찬한 것이다.

『資治通鑑』은 北宋의 司馬光이 周 威烈王 23년(B.C. 403)에서 시작하여 五代의 後周 世宗 顯德 6년(959)까지 모두 16왕조 1362년간의 史蹟을 編年體로 엮은 역사서이다. 모두 294권으로 편찬한 역사서로 司馬遷의 『史記』와 더불어 역사서의 명작이라고 일컬어진다.¹⁴⁶⁾ 『資治通鑑』은 하나의 한 王朝를 1紀로 하여 모두 16紀로 나누어 軍國大事와 君臣言行을 年月日에 따라 기록하였다. 『資治通鑑』에서 인용한 사료를 살펴보면 西漢의 사적은 『史記』와 『漢書』를 채용하고, 東漢은 范曄의 『後漢書』에서 열의 일고여덟을 채용하였으며, 魏·晉에서 隋代까지는 정사를 채용한 것이 열의 예닐곱이고, 唐은 正史를 채용한 것이 열의 다섯에 못미치며, 五代는 모두 薛居正的 『舊五代史』를 근거하였다.¹⁴⁷⁾ 그 밖에도 『資治通鑑』에 인용된 사료는 약 330여 종에 이른다고 하는데 모두 司馬光의 刪削과 纂修를 경유하였으므로 문체가 한사람의 손에서 나온 것과 같다고 하였다.¹⁴⁸⁾

『資治通鑑』은 帝紀와 列傳을 중심으로 기술하는 紀傳體에서 탈피하여, 시간의 흐름에 따라 일어난 사건을 종합적으로 기록하는 編年體로 기술하였다. 編年體 史書는 『春秋』로부터

143) 『譯註 通鑑節要』는 전통문화연구회 웹 사이트(<http://db.cyberseodang.or.kr>)에서 직접 확인할 수 있다.

144) 南宋 建州 崇安人, 字는 叔圭이다. 上庠에서 유학하여, 龔深과 함께 『주역』을 배워 이름이 났으나, 후에 은거하였다. 徽宗 政和 연간에 遺逸로 천거되어, 3번 초빙하였으나 끝내 벼슬에 나가지 않았다. 조정에서 少微先生이라는 號를 주었다. 『中國歷代人名大辭典 上』, 713쪽.

145) 『資治通鑑』의 번역서는 권중달 중앙대 명예교수가 역주에 착수한 지 14년만에 32권으로 도서출판 삼화에서 간행되었다.

146) 中砂明德(저)/강길중·김지영·장원철(역), 『우아함의 탄생』, 민음사, 2009, 100쪽.

147) 劉節(저)/辛太甲(역), 『中國史學史』, 서신원, 2000, 355쪽.

148) 이종동·조성을, 『중국사학사』, 혜안, 2009, 205쪽.

비롯되었는데 『史記』가 나오면서부터 후대의 正史는 대부분 『史記』의 편찬 체제인 紀傳體로 기술하는 것이 관행이 되었다. 그리하여 北宋 대에 이르면 正史만 하더라도 16종이나 되었으며 그 분량만 하더라도 1,600권에 달하는 방대한 양이 되었다. 그런데 紀傳體는 일종의 분류사이기 때문에 하나의 사건이 列傳, 書, 表 등에 중복되어 서술되어 있어서 자연스럽게 분량이 많아지게 되었다. 분량이 많다는 것은 역사서를 읽는데 큰 장애 요소가 된다.

이러한 문제를 해결하고자 司馬光은 역사를 紀傳體가 아닌 編年體로 기술하였다. 司馬光은 역사적 사실을 시간의 흐름에 따라 종합적으로 기록하는 編年體로 歷史를 기술함으로써 『史記』 이래 유행했던 紀傳體 역사 서술에서 編年體 史書의 전통을 부활시킨 것이다.¹⁴⁹⁾

이러한 『資治通鑑』이 과거에 지닌 영향력은 『史記』를 능가하였다. 그렇게 널리 읽히게 된 이유를 中砂明德은 다음과 같이 세 가지 요인으로 제시하였다. 첫째, 『資治通鑑』은 황제에게는 통치의 도움이 되는 거울이 되었고, 신하에게는 出處進退의 길잡이 역할을 했기 때문이다. 둘째, 중국의 通史를 제공하기 때문이다. 『史記』를 비롯한 대부분의 역사서가 하나의 왕조를 대상으로 역사를 기술하고 있는데 비해서 『資治通鑑』은 전국 시대부터 宋 이전까지의 역사를 통사적으로 다루고 있다. 수많은 正史를 모두 구비하고 읽는다는 것은 쉽지 않은 일이다. 그런 점에서 『資治通鑑』은 이전에 비해 훨씬 적은 분량으로 중국사를 이해할 수 있게 하였다. 셋째, 『資治通鑑』의 축약판의 영향력 때문이다. 『資治通鑑』이 비록 正史에 비하면 상대적으로 적은 분량이지만은 하나 294권은 여전히 부담스러운 분량이다. 그래서 등장한 것이 『通鑑節要』, 『資治通鑑綱目』 등의 折本이다. 이러한 通鑑 일족의 영향으로 『資治通鑑』 또한 오랫동안 사랑을 받아 왔다는 것이다.¹⁵⁰⁾

『資治通鑑』은 당시에 전해지던 正史의 총량에 비하면 그 분량이 많이 축소된 것이지만 그럼에도 불구하고 294권이라는 분량은 적지 않은 분량이라 사람들이 쉽게 읽을 수가 없었다. 司馬光조차도 이를 염려하여 연표 형식의 목록과 그 위에 『學要曆』이라는 번거로움과 간략함의 중간 정도 되는 ‘대사건 연표’를 만들어 어떻게든 전체를 살려 보려고 고심했다.

이러한 노력은 『資治通鑑』이 나온 이후에도 계속된다. 袁樞의 『通鑑紀事本末』과 江贊의 『通鑑節要』, 朱熹의 『資治通鑑綱目』 등의 축약본이 나온 것도 모두 이러한 중국사에 대해 좀 더 쉬운 접근 방법을 제공하려는 의도였다. 袁樞는 『通鑑紀事本末』를 지어, 編年體의 특성으로 인해 하나의 사건이 여기저기에 흩어져 있는 것을 사건별로 관련 기사를 한데 모음으로써 사건의 전말을 쉽게 파악할 수 있게 하였다. 江贊은 『資治通鑑』 294권을 50권으로 요약한 『通鑑節要』를 지었다. 朱熹는 『資治通鑑綱目』을 지어, 사건의 요점을 대문자인 綱으로 제시하고, 그것을 다시 소문자인 目으로 부연하는 형식을 취하였으며, 권수도 『資治通鑑』

149) 中砂明德(저)/강길중 외(역), 앞의 책(2009), 100쪽.

150) 中砂明德(저)/강길중 외(역), 앞의 책(2009), 100쪽.

』의 5분이 1 수준인 59권으로 줄었다.

『通鑑節要』는 北宋 시대의 江贄에 의해서 쓰여진 이후에 여러 번 修正과 潤色이 가해진 다음에 지금과 같은 형태의 책이 된 것으로 보인다. 그렇기 때문에 『通鑑節要』의 저술 시기가 北宋대임에도 불구하고 南宋대 朱熹가 지은 『資治通鑑綱目』의 영향을 받을 수 있었던 것이다.

먼저 『通鑑節要』의 권차를 살펴보면 50권 중 漢紀가 22권¹⁵¹⁾, 唐紀가 14권이나 되어 대부분의 역사를 漢과 唐 위주로 엮었으며, 六朝와 五代에 있어서는 큰 사건이나 열거하는 데 그치고 있다.¹⁵²⁾

표 17. 『資治通鑑』과 『通鑑節要』의 卷次

時代	時期	資治通鑑		通鑑節要	
		卷數	比率	卷數	比率
戰國時代	周紀	5권	3%	3권	6%
秦	秦紀	3권			
漢	漢紀	60권	20%	20권	40%
三國時代	魏紀	10권	3%	2권	4%
晉	晉紀	40권	14%	5권	10%
南北朝 時代	宋紀	16권	18%	3권	6%
	齊紀	10권			
	梁紀	22권			
	陳紀	10권			
隋	隋紀	8권	3%	1권	2%
唐	唐紀	81권	28%	14권	28%
五代十國	後梁紀	6권	10%	2권	4%
	後唐紀	8권			
	後晉紀	6권			
	後漢紀	4권			
	後周紀	5권			
		294권	100%	50권	100%

『大明一統志』에 의하면 『通鑑節要』의 저자인 江贄는 宋나라 政和 연간(1111~1117)에 宋 徽宗이 處士의 상징인 少微星이 나타난 것을 계기로 인재를 널리 구할 때 천거되었음에도 불구하고 조정의 부름에 응하지 않아 도리어 명성을 얻었고 후에 ‘少微先生’으로 불리게 되었다고 한다.¹⁵³⁾ 南宋 嘉熙 원년(1237년)에 쓰여진 江鎔의 『通鑑節要』 序文에 의하면 이 책을 朱子에게 질정하였고 그 후 南山主人 江淵이 다시 윤색을 가하여 여러 史書의 表·

151) 劉邦의 西漢과 光武帝의 東漢, 劉備가 건국한 蜀漢을 모두 합하여 22권이다. 『通鑑節要』는 삼국 시대를 蜀을 정통으로 두고 나머지 두 나라를 별도로 구분하지 않았다.

152) 成百曉, 『譯註 通鑑節要』1, 전통문화연구회, 2007, 13쪽.

153) 『大明一統志』 권 76.

志·序·贊과 유명한 분들의 論評 및 音註를 붙인 것으로 되어 있다.¹⁵⁴⁾ 그런데 애초의 『通鑑節要』는 『資治通鑑』을 따라 魏나라를 정통으로 편찬하였으나 후대에 朱子の 『資治通鑑綱目』의 체제를 따라 이를 바꾸었으며, 新나라의 王莽과 漢나라의 呂后, 唐나라의 則天武后에 대해서도 정통성이 결여되었다 하여 연도를 표기할 적에 간지 아래에 小字雙行으로 표기하였다. 뿐만 아니라 周나라가 망한 뒤에 곧바로 秦紀로 쓰지 않고 東周君 7년을 더 넣고 東周가 망한 뒤에야 비로소 周나라가 망한 것으로 표기하였다.

그런데 이러한 『通鑑節要』는 宋版은 커녕 元版의 존재도 확인되지 않고 있다. 원나라 시대에 『通鑑節要』가 몽골어로 번역이 이루어지지만 그 번역의 대본이 少微先生의 것이라는 확증은 없다.¹⁵⁵⁾ 『通鑑節要』가 본격적으로 보급되는 것은 明代의 劉劄¹⁵⁶⁾에 의해서이다. 劉劄은 출판 교섭을 위해 고향인 江西에서 떠나 建陽을 찾은 王逢을 만나게 된다. 王逢으로부터 『通鑑節要』가 역사 학습의 지름길이며 朱子の 『資治通鑑綱目』의 입문서로도 적당하다는 말을 듣고 王逢에게 그 책을 보여달라고 하였다. 그러자 王逢은 그 책에 대한 자신의 주석인 『通鑑節要釋義』를 보여 주었다. 劉劄은 王逢의 註釋을 본문에 끼워 넣어, 민간의 출판사에서 간행하기에 이르렀다.¹⁵⁷⁾ 명나라 武宗도 이 책을 좋아하여 司禮監에 책의 重刊을 명하고 황제가 직접 序文까지 하사했다고 한다. 이렇듯 『通鑑節要』는 국가에서도 보증을 받았다.

『通鑑節要』은 朝鮮에 전해지고 나서 활자로 간행되고 日本에도 전해졌다. 그뿐만 아니라 베트남에도 전해져 간행되기도 하였다.¹⁵⁸⁾ 『通鑑節要』가 우리나라에 전해진 것은 확실하지는 않다. 李詹¹⁵⁹⁾의 『雙梅堂先生篋藏文集』 권25에 의하면 1381년(禡王 7년)에 朴某가 河崙에게서 『少微通鑑』 한 질을 빌려서 重刊을 하였다는 기록이 나와 있다.

辛酉春。今鷄林府尹朴公某。以江州兵馬使兼牧晉陽。戎務既克修舉。因在心史典。遂求通鑑于

154) 成百曉, 『譯註 通鑑節要』1, 전통문화연구회, 2006, 13쪽. 江鎔의 『通鑑節要』 서문에 의하면

『通鑑節要』라는 책이 少微先生 江氏의 家塾에 있었는데 建寧公 江默이 朱熹의 문하에서 受學할 적에 일찍이 이 책을 가지고 질정하니, 朱先生은 크게 감탄하고 칭찬하는데 이로부터 士友들이 다투어 서로 전하여 기록해서 더욱더 소중하게 여겼다고 한다.

155) 中砂明德(저)/강길중 외(역), 위의 책(2009), 116쪽.

156) 명나라 유학자로 建陽(福建)인. 字는 用章이며, 『通鑑節要』와 『十八史略』을 간행하였다.

157) 中砂明德(저)/강길중 외(역), 위의 책(2009), 114~115쪽.

158) 『우아함의 탄생』 p117에 의하면 파리 국립도서관에 소장된 한적 목록을 살펴보면 제1장 2절의 「연대기」 항목에 몇 종류의 『通鑑節要』가 언급되어 있는데 그 중에 1840년에 간행된 安南版도 들어 있다고 한다.

159) 李詹 : 1345년(충목왕 1)~1405년(태종 5). 고려 말 조선 초의 문신. 本貫은 新平. 字는 中叔, 號는 雙梅堂. 조부는 寶文閣提學 李達尊이고, 부는 贈參贊議政府事 李熙祥이다. (온라인 참조: 이침, 한국민족문화대백과사전, 한국학중앙연구원, http://encykorea.aks.ac.kr/Contents/Index?contents_id=E0046240).

典理判書河公崙。得少微一本。募工重刊。時以軍國事殷。或有沮公意者。公不聽。適安君景公，呂君充禔相繼爲按部。率能有以左右。事既半。公見代而歸。今通判江陵崔公云嗣與牧伯林公子安。克續厥功。能事既畢。嗚呼。數君子之用心。東方人士感受其賜。豈不休哉。豈不幸哉。謹跋。160)

이 내용을 통해서 『通鑑節要』는 최소한 高麗 시대에 이미 우리나라에 유입된 것을 확인할 수 있다. 『通鑑節要』가 重刊된 것은 『通鑑節要』가 간행된 1237년으로부터 140여년이 지난 뒤의 일이다.

『通鑑節要』가 조선 시대 지식인에게 끼친 영향은 적지 않다. 조선 시대에 사대부들은 한문 학습 과정에서 『通鑑節要』를 통해서 문리를 터득하였다. 이러한 사실은 李珣가 李成春에게 문리가 나지 않았으므로 『通鑑』을 읽을 것을 권하는데서도 알 수 있다. 『通鑑節要』는 조선 시대로부터 오늘날까지 널리 읽혔음에도 불구하고 여러 학자들에게 많은 비판을 받아왔다. 『通鑑節要』를 신랄하게 비판한 학자는 丁若鏞이다. 『與猶堂全書』 제1집 通鑑節要評에 다음과 같은 내용이 보인다.

“이 책은 司馬溫公의 『資治通鑑』을 藍本으로 삼았으나 그 義例는 도리어 朱子의 『資治通鑑綱目』을 기준하였다. 그리하여 三國에 있어 正統은 蜀漢에 주고 記事는 曹魏를 위주로 하여 주객이 바뀌고 王賊이 전도됨으로써 의리에 온당치 못하다. 기타 연월의 착오와 사실의 와전은 이루 헤아릴 수 없어 논란할 필요조차 없다.”¹⁶¹⁾

洪翰周는 『智水拈筆』에서 『通鑑節要』에 대해서 한층 더 신랄하게 비판한다.

江贄라는 자는 宋代의 한 글방 훈장일뿐이다. 일찍이 문학으로 당시 사람들에게 이름이 알려지지 않았었고, 宋·元·明·淸 이래 筆錄과 野乘에 한 사람도 江贄를 일컬은 이가 없다. 다만 그가 엮은 『通鑑節要』 15권이 민간에서 판각되어 세상에 유행했기 때문에 비로소 江贄라는 이가 있는 줄을 알지만, 그러나 그 책은 溫公의 『資治通鑑』을 가지고 초록하고 節錄하여 권질을 줄여서 아이들을 가르치기에 편하도록 한 것일 뿐이다. 抄錄하고 節錄하는 일이란 번다한 것을 깎아 요점을 따오기만 하면 되는 것인데, 이 책은 애초 원칙이 없이 다만 깎아서 줄이기만 힘썼다. 그래서 모두 구절을 가르고 찢어 문리가 이루어지지 않으니, 이 어찌 역사서라 할 수 있겠는가. 명나라 초에 우리나라 사신이 燕京에 갔다가 우연히 서점에서 이 책을 파는 것을 보고 권수가 적고 값이 싼 점 때문에 사왔다. 곧 간행하여 지금은 온 나라 안에 퍼져서 무릇 처음 배우는 아이들은 귀전을 막론하고 사람사람이 외고 익혀서 천지 사이에서 바뀌지 않을 위대한 저술로 알고, 『通鑑綱目』은 도리어 이 책보다 뒤로 여긴다. 촌구석 서당 훈장이 우리나라에서

160) 『雙梅堂先生篋藏文集』, 卷之二十五, 跋類(온라인 참조 : 한국고전종합DB, 한국고전번역원,

http://db.itkc.or.kr/inLink?DCI=ITKC_MO_0031A_0070_020_0090_2003_A006_XML).

161) 第一集詩文集第二十二卷 文集 雜評, 한국문집총간, 한국고전번역원(온라인 참조 :

http://db.itkc.or.kr/dir/item?itemId=MO#dir/node?grpId=&itemId=MO&gubun=book&depth=5&cate1=Z&cate2=&dataGubun=최종정보&dataId=ITKC_MO_0597A_0220_040_0030)

이토록 득세할 줄을 어찌 알았으랴.¹⁶²⁾

洪翰周의 비평은 내용을 줄이는 데 원칙이 없어 문리가 통하지 않는다는 것이 가장 큰 지적임을 알 수 있다. 뒤에 다시 언급하겠지만 『通鑑節要』는 내용을 극도로 압축하다보니 『資治通鑑』에서 여러 날에 걸쳐 발생한 사건을 하나의 문장으로 압축해 놓은 것이 여러 곳에서 발견된다. 엄밀한 정확성을 요구하는 史書로서는 큰 결함이 있다고 할 수 있다.

『通鑑節要』의 특징을 살펴보면 첫째, 劉劭의 주장대로 統紀를 바로잡았다는 점을 들 수 있다. 그 예를 들면 江贄는 南宮精一의 史論을 인용하여 周나라의 정통이 赧王이 아닌 東周君까지 이어진 것으로 보았으며¹⁶³⁾, 秦始皇帝를 呂不韋의 혈통으로¹⁶⁴⁾ 보고 秦나라를 「秦紀」와 「後秦紀」로 나누어 기술하였으며,¹⁶⁵⁾ 漢高祖 劉邦이 건국한 漢을 「漢紀」로, 光武帝가 건국한 「後漢」을 「東漢紀」로, 劉備가 건국한 蜀漢을 「後漢紀」로 기술하였다. 둘째, 『資治通鑑』을 節錄하면서 제대로 하지 못한 부분과 오류가 많다는 점이다. 『資治通鑑』을 50권으로 줄이면서 서술이 자연스럽지 못한 부분이 많으며, 『資治通鑑』에 없는 記事를 추가한 경우도 있다. 예를 들어 秦始皇帝가 東南 지방에 天子의 기운이 있다고 하여 직접 巡幸하여 그 기운을 누를 때 漢高祖가 芒縣과 碭縣의 산과 늪 사이에 숨어 있었는데, 그럴 때마다 呂后가 항상 漢高祖를 찾아내곤 해서 그 이유를 물으니 呂后가 “季가 머무는 곳 위에는 항상 구름 기운이 있기 때문에 따라가면 항상 季를 찾아냅니다.”라고 한 부분이 그렇다.¹⁶⁶⁾ 여기서 구름 기운이 있다는 말은 『史記』에는 있으나 『資治通鑑』에는 빠진 내용인데 이를 되살려 놓은 것이다. 셋째, 史書가 갖추어야 할 기본적인 요건이라고 할 연도 기록의 정확성이 너무 떨어진다는 점이다. 넷째, 『通鑑節要』는 『資治通鑑』을 축약하면서 서로 다른 날짜의 기사를 하나로 묶어서 여러 날에 걸쳐서 일어난 사건이 마치 동일한 시점에 발생한 것으로 혼동할 수 있게 하였다.

이렇듯 결함이 많은 『通鑑節要』를 대상 자료로 선택한 이유는 다음과 같다. 우선 『通鑑節要』는 조선 시대 전 기간에 걸쳐 중국 역사에 대한 지식뿐 아니라 여러 학문에 걸쳐 수많은 정보를 제공해 온 기본 교재로써 역할을 하였기 때문에 조선 시대 사대부의 역사 인식

162) 洪翰周, 『智水拈筆』, 김윤조, 「조선 후기 지식인들의 『通鑑節要』에 대한 비판적 인식의 양상과 의미」, 『한문학보』, 5, 2001, 258쪽을 재인용.

163) 『通鑑節要』는 周赧王이 사망했음에도 불구하고 東周君이 있기 때문에 周 왕조가 끝나지 않았다고 보고 7년 간의 東周君 시대를 설정하였다. 이는 『資治通鑑』에서는 周나라를 赧王에서 끝내고 그 이후는 「秦紀」로 기술한 것과 다르다.

164) 『史記』 「呂不韋列傳」에는 秦始皇의 생부가 呂不韋로 기록되어 있다.

165) 中砂明德(저)/강길중 외(역), 위의 책(2009), 118쪽.

166) 『通鑑節要』 卷之三 後秦紀 [壬辰]元年, 通鑑節要(온라인 참조 :

http://db.cyberseodang.or.kr/front/alphaList/BookMain.do?mId=m02&bnCode=jti_2b0101&titleId=C90, 고전종합DB, 전통문화연구회).

을 이해하기 위해서는 『通鑑節要』를 읽어야 한다. 『通鑑節要』는 조선 시대 각종 문집에도 별도의 표시 없이 술하게 인용되었으며 『朝鮮王朝實錄』 등에서도 빈번하게 고사가 인용되었다. 위에서 언급한 『通鑑節要』에 대한 수많은 비판은 역설적으로 조선 시대에 『通鑑節要』가 널리 읽혔음을 보여주는 반증이기도 한다. 다음으로 『通鑑節要』는 『資治通鑑』에 비해 내용이 매우 압축적이라는 점을 고려하였다. 즉 『資治通鑑』이 번다한 것에 비해 핵심적인 내용이 주로 담겨 있기 때문에 오히려 사건을 중심으로 기술할 때 더 유리하다. 또한 『通鑑節要』는 司馬光의 『資治通鑑』, 朱熹의 『思政殿訓義 資治通鑑綱目』 등 외부 데이터와 연계시킬 수 있는 최적의 조건을 갖추고 있기 때문이다. 본 논문은 역사서에 나타나는 행위를 중심으로 역사적 지식의 유형을 밝히고 그 표현 방법을 찾고자 하는 것이 목적이기 때문에 이 점에서 『通鑑節要』는 최적의 조건을 갖추고 있다고 할 수 있다.

朝鮮에서 가장 널리 유통되었던 판본은 江贊의 판본을 저본으로 하여 위에서 언급한 명나라 초에 간행된 劉剗增校本이다. 劉剗本 『通鑑節要』는 庚子字로 인쇄한 鄭麟趾 수택본이 성암고서박물관에 2책, 산기문고에 1책이 남아 있다고 한다.¹⁶⁷⁾

본 연구에서 활용한 『通鑑節要』 텍스트는 전통문화연구회에서 9책으로 간행한 『譯註 通鑑節要』이다.¹⁶⁸⁾ 우선 정확한 판본을 알기 위해 『譯註 通鑑節要』의 凡例를 살펴본다.

2. 本書는 가장 善本으로 보이는 甲寅字本 『少微家塾點校附音通鑑節要』¹⁶⁹⁾를 底本으로 하되 木版本 『少微家塾點校附音通鑑節要』¹⁷⁰⁾를 참고하였다. 이 책은 眉山 史炤가 音釋하고 鄱陽 王逢이 輯義하고 京兆 劉剗이 增校한 것이다.

3. 本書는 甲寅字本 『少微家塾點校附音通鑑節要』를 底本으로 하되 현재 春坊本 『通鑑節要』가 流行되고 있음을 감안하여 溫公의 史評은 本文과 같이 大字로 표기하였으며 기타 史論은 글자 크기를 약간 줄였다. 그리고 底本の 史評 외에 『二十史略』의 史評을 추가하여 『史略 史

167) 허태용, 「17세기 후반 正統論의 강화와 『資治通鑑節要』의 보급」, 『한국사학사학보』, 제2권, 2000, 17쪽.(성암고서박물관은 현재 문을 닫은 상태이다. 산기문고는 인사동에 通文館을 운영하는 山氣李謙魯의 개인 문고이다.)

168) 成百曉, 『譯註 通鑑節要』1, 전통문화연구회, 2005.
 成百曉, 『譯註 通鑑節要』2, 전통문화연구회, 2005.
 成百曉, 『譯註 通鑑節要』3, 전통문화연구회, 2006.
 成百曉, 『譯註 通鑑節要』4, 전통문화연구회, 2006.
 成百曉, 『譯註 通鑑節要』5, 전통문화연구회, 2007.
 成百曉, 『譯註 通鑑節要』6, 전통문화연구회, 2008.
 成百曉, 『譯註 通鑑節要』7, 전통문화연구회, 2008.
 成百曉, 『譯註 通鑑節要』8, 전통문화연구회, 2009.
 成百曉, 『譯註 通鑑節要』9, 전통문화연구회, 2009.

169) 國立中央圖書館 所藏本, 刊年未詳.

170) 高麗大學校 圖書館 所藏本 및 서울大學校 奎章閣 所藏本.

評]이라고 표시하였다.¹⁷¹⁾

다음으로 『通鑑節要』의 체제를 살펴본다. 대상 자료로 삼은 전통문화연구회에서 간행한 『譯註 通鑑節要』의 범례를 살펴보면 “甲寅字本을 底本으로 하되 木版本의 頭註와 『通鑑要解』¹⁷²⁾를 추가하였으며, 현재 春坊本이 流行함을 감안하여 濫公의 논평은 本文과 같이 大字로 표기하였으나 기타의 史論은 글자 크기를 약간 줄였다. 그리고 글자의 간단한 훈이나 음은 모두 신지 않았으며, 小字雙行으로 중간에 실려 있는 주석은 단락이 끝나는 곳에 붙였음을 밝혀둔다.” 라고 하였다. 여기서 甲寅字는 戊申字의 오류로 보인다. 위의 내용을 볼 때 『譯註 通鑑節要』는 江贇의 저작에 劉劄이 주석을 추가하였으며, 현대에 국역본을 내면서, 『通鑑要解』와 二十史略 등을 추가한 새로운 형태의 저작이라 할 수 있다. 이를 정리하면 다음과 같다.

표 18. 『譯註 通鑑節要』의 체제

구분	내용	
註釋	頭註	광곽 위에 있는 주석
	原註	별도의 표시가 없이 小字雙行으로 된 주석
	釋義	陰刻으로 釋義라고 표시된 小字雙行 주석
	附註	卷마다 끝에 연도별로 붙인 주석
本文	本文	通鑑節要 본문
史論	司馬光 史評	司馬光의 사론
	기타 史論	별도의 표시가 없는 註는 史炤가 단 것 輯義(釋義)는 王逢이 단 것 新增 등의 표시가 있는 史論은 劉劄
	史略 史評	二十史略의 史評을 추가

다음으로 『譯註 通鑑節要』의¹⁷³⁾ 데이터 구조에 대해서 간단히 살펴본다.¹⁷⁴⁾ 최종 데이터

171) 成百曉, 『譯註 通鑑節要』1, 전통문화연구회, 2007, 29쪽.

172) 成百曉, 『역주 통감절요』1, 전통문화연구회, 2007, 16쪽. 『譯註 通鑑節要』 해제에 따르면 『通鑑要解』의 정식 명칭은 『通鑑五十篇詳節要解』로 九淵禪師가 편찬하였고 李在璣가 序文을 지었다.

173) 『譯註 通鑑節要』는 교육부의 예산 지원으로 수행하는 고전문헌 국역지원사업 중 하나인 ‘동양고전종합DB 구축 사업’의 결과로 구축된 자료이다. 동양고전종합DB 구축 사업은 2010년에 처음 시작되어 2017년 현재까지 연차 사업으로 계속되고 있으며 사업의 가장 큰 역할은 동양고전을 정보화하여 일반인들이 쉽게 접근할 수 있게 하였다는 점이다. 전통문화연구회에서는 고전번역에 필요한 사전 어휘 수집 및 보급을 위한 고전어휘 db 구축 사업도 2013년부터 지속적으로 진행해 오고 있다.

구조를 살펴보면 원문과 번역문이 단락을 기본 단위로 하여 반복되는 형태이며 단락 엘리먼트 내부에는 각주와 원주가 있으며 각주의 내용은 별도의 주석묶음을 엘리먼트로 분리하였다.

표 19. 『通鑑節要』의 데이터 구조

```

<원문 식별자="ID:W3" lang="chi">
<단락 indent="0">○ 是歲에 突厥餘黨이 入寇并州어늘 薛仁貴將兵擊之할새 虜問唐大將爲誰오 應之曰 薛仁貴로라 虜曰 吾聞仁貴流象州하야 死久矣어늘 何以給我오 仁貴免<각주 식별자="ID:AW2661">靑</각주>示之面하니 虜相顧失色하야 下馬列拜하고 稍稍引去어늘 仁貴因奮擊하야 大破之하다</단락>
</원문>

<번역문 식별자="ID:W3_T" lang="kor">
<단락 indent="0">이해에 突厥의 잔당이 并州를 침략하였는데, 薛仁貴가 장차 그들을 치려 할 적에 오랑캐가 묻기를 “唐나라 대장이 누구인가?” 하자, 대답하기를 “薛仁貴이다.” 하니, 오랑캐가 말하기를 “내 들으니 薛仁貴는 象州로 유배가서 죽은 지가 오래되었는데 어찌 우리를 속이는가.” 하였다. 薛仁貴가 투구를 벗고 얼굴을 보여주니, 오랑캐가 보고 아연실색하여 말에서 내려 늘어서서 절을 하고는 이윽고 병사들을 데리고 떠나갔다. 薛仁貴가 이 틈을 타고 진격해서 이들을 크게 격파하였다.</단락>
</번역문>

<주석묶음>
<각주 식별자="ID:AW2661"><주석명>靑</주석명><주석내용>주</주석내용></각주>
</주석묶음>

```

전통문화연구회에서 간행한 『譯註 通鑑節要』의 자료량은 다음과 같다. 대략 원문과 번역문을 합쳐서 270만 자이며 어절로 따지면 57만 어절이 된다. 원문과 번역문, 각주로 구분하였다.

표 20. 『譯註 通鑑節要』의 자료량

구분	글자수	어절수	비고
원문	872,179	122,733	
번역문	1,716,971	401,715	
각주	193,532	47,161	
합계	2,782,682	571,609	

『通鑑節要』를 온톨로지로 구축하기 위해서는 역사서라는 특징 때문에 시간이란 요소가

174) 동양고전종합DB, 전통문화연구회, <http://db.cyberseodang.or.kr/front/main/main.do>

매우 중요하다. 이와 관련하여 『通鑑節要』에서 시간과 관련하여 주의해야 할 점을 살펴보자. 『史記』를 비롯한 紀傳體 역사 기술이 동일한 사건이 여러 곳에 중복되어 나타나는 문제점으로 인해 司馬光은 歷史를 編年體로 기술하였다. 그런데 編年體로 역사를 기술한다고 해서 모든 문제가 해결되는 것은 아니다. 오히려 紀傳體에 없었던 새로운 문제가 생기기도 한다. 예를 들어 역사적 事件이라는 것이 어느 하루에 이루어지는 것이 아니고 여러 날 혹은 몇 년에 걸쳐서 진행되기도 하기 때문에 이러한 사건을 날짜 순으로 기록한다면 전체적인 사건을 이해하는데 문제가 될 수 있다. 司馬光은 이러한 문제를 해결하기 위해 사건의 중심이 되는 날짜에 원인과 결과를 같이 기술하였다.¹⁷⁵⁾ 그러나 다른 날짜에 발생한 사건을 그 날 날짜에 기록한다는 것은 자칫하면 그 날 일어난 사건으로 오해할 수 있는 여지가 있다. 이러한 문제를 해결하기 위해 司馬光은 ‘初’자를 사용함으로써 사건이 발생한 시점이 그날이 아닌 과거라는 것을 표현하였다. 이렇게 시점이 서로 다름을 표시하는 것은 ‘初’ 뿐만 아니라 ‘嘗, 先是, 少’ 등이 있다. 그리고 특별히 이러한 용어를 사용하지 않더라도 문맥 상에서 자연스럽게 드러나는 경우도 있다. 이러한 사례는 우리말로 번역할 때 ‘처음에, 처음으로, 이미, 이때, 있을 때에, 곧, 일찍이, 보다 앞서, 지난날에, 얼마 후, 초에, 여러 해가 지나도, 이보다 앞서, 뒤에, 태어나면서부터 곧바로, 곧이어, 곧장, 젊어서부터’ 등 문맥에 따라 다양한 형태로 번역이 된다. 이러한 시간의 기술 방법은 『通鑑節要』에서도 당연히 그대로 유지되고 있다.

이러한 시점의 차이를 언급하는 이유는 사건을 기술할 때 반드시 시간 정보를 기술하는 시점과 비교하여 같은 시기인지 확인해야 하기 때문이다.

『通鑑節要』에는 시간과 관련하여 새로운 문제를 안고 있다. 『通鑑節要』는 『資治通鑑』을 축약하면서 서로 다른 날짜의 기사를 하나로 묶어서 여러 날에 걸쳐서 일어난 사건이 마치 동일한 시점에 발생한 것으로 혼동할 수 있게 하였다. 예를 들어 『通鑑節要』 권49 五代紀 後梁紀 太祖皇帝 [丁卯]唐天祐四年 기사에 “봄 3월에 唐나라 昭宣帝가 御札을 내려 梁나라에 황제의 지위를 禪讓하자, 梁王이 이름을 晁으로 고치고 황제에 즉위하여 국호를 梁이라고 하고 昭宣帝를 받들어 濟陰王으로 삼았다.”¹⁷⁶⁾고 되어 있는데, 『資治通鑑』에 따르면 唐나라 昭宣帝가 어찰을 내려 後梁에 皇帝의 자리를 禪讓한 것은 907년 1월 27일의 일이고, 朱全忠이 이름을 朱晁으로 고친 것은 907년 4월 16일의 일이다. 황제에 즉위한 것은 907년 4월 18일이며, 국호를 梁이라고 하고 昭宣帝를 濟陰王으로 삼은 것은 907년 4월 22일이다. 그런데

175) 권중달, 『『자치통감』의 사학사적 의미』, 『한국사학사학보』 21호, 2010, 207~208쪽.

176) 通鑑節要 卷之四十九 五代紀 後梁紀 太祖皇帝 [丁卯]唐天祐四年, 通鑑節要(온라인 참조 :

동양고전종합DB, 전통문화연구원,

http://db.cyberseodang.or.kr/front/alphaList/BookMain.do?mId=m02&bnCode=jti_2b0109&titleId=C977).

『通鑑節要』에는 907년 봄 3월 기사에 이 4일의 기록을 묶어서 마치 하나의 기사처럼 기술하였다. 禪讓한 것과 濟陰王으로 삼은 내용 사이의 시간은 거의 3개월이나 떨어져 있다. 또 다른 기사로 『通鑑節要』 권49 五代紀 後唐紀 莊宗[癸未]의 기사를 살펴보면 “겨울 10월에 唐主가 大軍을 거느리고 黃河를 건너 鄆州에 이르러서 梁나라 군대를 만나 一戰에 패퇴시키고, 추격하여 中都에 이르러서 그 성을 포위하니, 성에 수비가 없었다. 잠시 후에 梁나라 군대가 포위를 뚫고 나오자, 王彥章을 사로잡아 목을 베었다.”¹⁷⁷⁾로 되어 있다. 이 내용을 『資治通鑑』에서 확인하면 後唐의 李存勖이 黃河를 건넌 것은 923년 10월 2일의 일이며, 鄆州에 도착한 것은 10월 3일의 일이다. 梁나라의 군대를 一戰에 패퇴시키고 王彥章을 사로잡은 것은 10월 4일의 일이며, 王彥章의 목을 베는 것은 10월 5일의 일이다. 4일 간의 사건이 하나의 문장으로 요약된 것이다.

『通鑑節要』에는 이와같이 극단적인 기사 요약이 빈번하다. 이러한 문제로 인해 『通鑑節要』가 史料로서의 가치가 그리 크지 않지만 역사의 대략을 이해하는 데는 많은 도움이 된다. 다시 말해 자세한 역사적 지식을 전달한다는 측면에서는 부족한 점이 많으나 대략적인 역사의 흐름을 파악하기는 매우 유용하며, 이러한 점이 『通鑑節要』가 오랫동안 읽혀 왔던 이유 중의 하나인 것으로 생각된다.

이러한 시간의 표현은 역사적 지식을 온톨로지표현할 때 주의해야 한다. 즉 ‘初’字 등으로 표현한 사건의 구체적인 날짜를 확인하고, 만약 확인할 수 없으면 비워두어야 한다. 그리고 『通鑑節要』에는 구체적인 날짜까지 기록되는 일이 있다. 『通鑑節要』 권49 五代紀 後唐紀 莊宗 [乙酉]後唐同光三年 기사를 살펴보면 “戊申日(6월 18일)에 大軍이 서쪽으로 출동하여 散關으로 들어가서 행군 속도를 배가하여 전진하니, 王承捷이 鳳州·興州·文州·扶州 네 州의 印信 및 武興節度使의 印信과 旌節을 가지고 大軍을 맞이하여 항복하였으며, 그 나머지 城鎮도 모두 소문만 듣고 歸附하였다.”¹⁷⁸⁾라고 하였는데 이 기사에서 大軍이 서쪽으로 출동한 것은 925년 6월 18일이며, 散關으로 들어간 것은 10월 18일이며, 네 주를 점령한 것은 10월 19일의 일이다. 즉 『通鑑節要』에서 구체적인 날짜의 표현이 온전하게 그날 발생한 사건이 아닐 수도 있다는 것이다. 이런 점 때문에 『通鑑節要』에서 구체적인 날짜가 있어도

177) 通鑑節要 卷之四十九 五代紀 後唐紀 莊宗 [癸未]岐稱唐天祐二十年, 通鑑節要(온라인 참조 : 동양고전종합DB, 전통문화연구회, http://db.cyberseodang.or.kr/front/alphaList/BookMain.do?mId=m02&bnCode=jti_2b0109&titleId=C992).

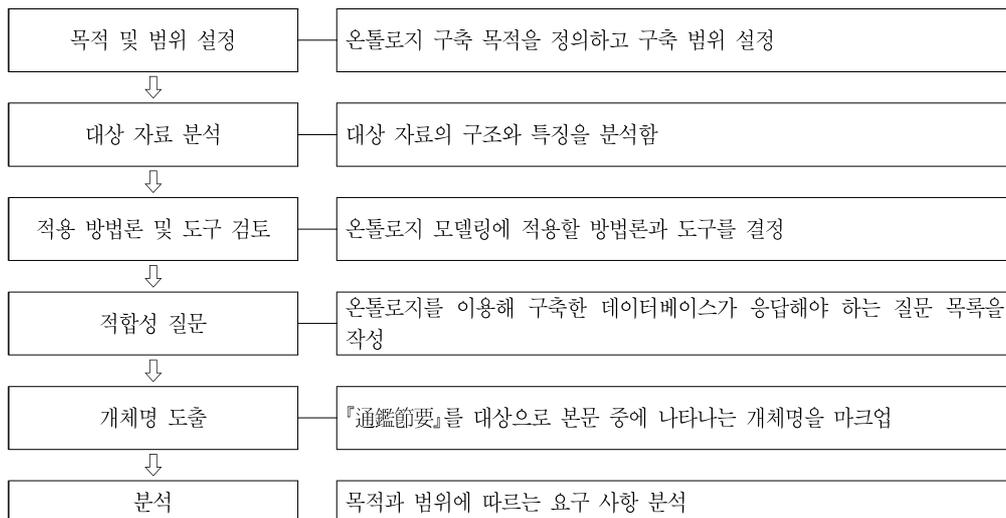
178) 通鑑節要 卷之四十九 五代紀 後唐紀 莊宗 [乙酉]後唐同光三年, 通鑑節要(온라인 참조 : 동양고전종합DB, 전통문화연구회, http://db.cyberseodang.or.kr/front/alphaList/BookMain.do?mId=m02&bnCode=jti_2b0109&titleId=C995).

구체적으로 표현하지는 않았다. 다만 온톨로지 설계에서는 년, 월, 일, 윤달 여부를 표현할 수 있도록 하였다.

2. 온톨로지 설계 절차

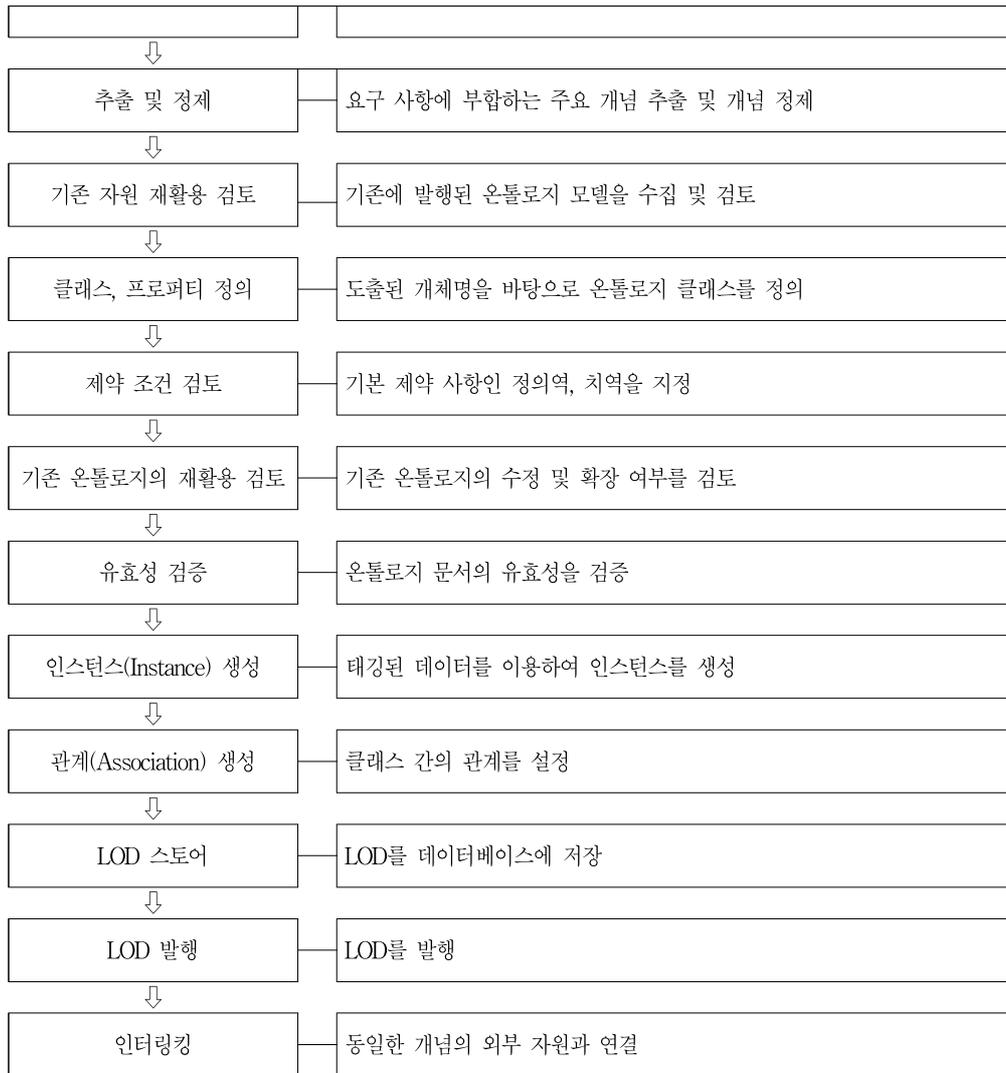
온톨로지 설계 방법은 일반적으로 대상 자원을 클래스로 범주화하고, 각각의 클래스에 속하는 개체들이 공통의 속성을 갖도록 하고, 그 개체들이 다른 개체들과 맺는 관계를 명시적으로 기술하는 것이다.¹⁷⁹⁾ 그러나 실제로 온톨로지를 설계하고 구축하기 위해서는 적절한 방법론을 선택하는 것이 좋다. 온톨로지 구축에 사용하는 방법론은 Ontology Development 101¹⁸⁰⁾, EOE(Evolving Ontology Engineering) 등이 있다. 이 중에 본 연구에서는 Ontology Development 101에서 제시하고 있는 구체적인 클래스 계층의 정의, 클래스와 클래스 명의 표현 방법 등을 참조로 하고, 절차는 EOE 방법론을 기본으로 하여 본 연구에 적합하도록 테일러링하여 적용하였다. EOE 방법론은 실제 구현과 모델링을 순환적으로 하는 방법으로 구체적인 구현을 통해서 시행착오를 수정해 나갈 수 있다는 점에서 유용하다고 할 수 있다. 온톨로지 구축은 다음과 같이 목적 및 범위 설정으로부터 최종적으로 LOD를 발행하는 단계까지 진행된다.

표 21. 구축 절차



179) 김현 외, 앞의 책(2016), 164쪽.

180) 박지영, 「온톨로지 개발을 위한 일반 지침(Ontology Development 101: A Guide to Creating Your First Ontology)」을 참조하면 된다.



가. 목적 및 범위

온톨로지 모델을 설계하기 위해서는 분명한 목적과 범위를 정한 후에 진행하는 것이 좋으므로 본 연구에서는 아래와 같은 ‘요구사항 정의서’를 만들어서 범위와 방법을 설정하였다.

표 22. 요구사항 정의서

구분	내용
목적	<ul style="list-style-type: none"> 역사적 지식을 LOD와 지식 관계망을 이용하여 연계 및 통합 LOD로 발행하여 외부에서도 자유롭게 활용할 수 있게 구현 원문과 번역문에 사용된 어휘 간 연관 관계를 표현함 지식 관계망을 통해 기존 LOD가 가지는 한계를 극복하고 한층 더 정제된 지식을 제공
범위	<ul style="list-style-type: none"> 역사서에 출현하는 본문 내용을 기술 역사적 지식을 행위나 상태를 중심으로 기술 원문과 번역문에 사용된 어휘를 양방향으로 참조할 수 있는 온톨로지 설계 역사서에 출현하는 문맥 노트로부터 새로운 정보로 이동할 수 있는 지식 관계망의 설계
특징	<ul style="list-style-type: none"> 역사적 사실을 의미 전달 위주로 기술 메타데이터가 아닌 문장을 표현할 수 있는 온톨로지 모델 설계 및 구현 사건 등의 정제된 지식을 기반으로 어휘 간 의미 및 내용적인 관계망을 구성
표현 언어	<ul style="list-style-type: none"> OWL DL, RDF, RDF Schema
기존 온톨로지 재활용	<ul style="list-style-type: none"> 클래스의 외연이 완전히 동일하나? 부분적으로 재활용할 수 있는가? Schema.org, DC, 한국사 LOD, 국립중앙도서관의 온톨로지 모델, CIDOC 등을 검토

나. 온톨로지 설계

1) 온톨로지 설계 원칙

가) 클래스와 인스턴스

온톨로지 설계를 위해서 제일 먼저 할 일은 정보화 하고자 하는 지식 세계에 어떠한 지식 요소들이 있는지 탐색하고, 그 성격을 분석하는 것이다.¹⁸¹⁾ 본 연구에서는 개체명의 구체적인 출현 상태와 분포 현황을 파악하기 위해 『通鑑節要』의 원문과 번역문에 개체명을 수작업으로 직접 마크업하고, 마크업된 결과를 분석하였다. 개체명이란 문서에서 나타나는 명사나 숫자 표현과 같은 문서의 고유한 성질을 표현하는 개체를 말한다. 개체명은 인명, 지명, 기관명 등의 이름 표현과 날짜나 시간과 같은 시간 표현, 금액이나 퍼센트 같은 수치 표현으로 구분할 수 있다.¹⁸²⁾ 이러한 개체명은 온톨로지에서는 대부분 클래스로 이용할 수

181) 김현 외, 앞의 책(2016), 166쪽.

있다. 개체명은 문맥 정보(인물, 장소, 사건, 개념 등)로 부르기도 한다. 이러한 문맥 요소는 이용자들에게 다양한 콘텐츠 접근점을 제공할 수 있다. 이는 결국 역사적 지식에 대한 접근성을 높이고, 그 내용에 대해 알고자 하는 동기를 유발하며, 이용자들이 하여금 그 역사와 관련된 다양한 지식을 획득할 수 있게 하는 효과를 발휘하게 한다.¹⁸³⁾

『웹 온톨로지 개발 지침』¹⁸⁴⁾에 따르면 클래스의 정의를 “우리가 사용하려 하는 두 개 이상의 속성과 두 개 이상의 개체를 지닌 동질성의 의미를 가진 독립적인 집합”이라고 했다. 여기서 “우리가 사용하려 하는”은 구축하고자 하는 분야의 목적과 범위, 전략에 따라서 달라질 수 있다는 의미이다. “두 개 이상의 속성과 두 개 이상의 개체”라는 것은 정의하고자 하는 클래스의 구성 요소가 두 개 이상의 개체를 가져야 한다는 의미이다. “동질성을 가진 집합”은 집합의 의미 부여에 있어 동질성을 갖고 있어야 한다는 의미이며 동질성이라는 것은 곧 속성이 같다는 것을 의미한다. 여기서 속성은 본질 속성을 말한다. 본질 속성은 어떤 개체의 아이덴티티를 결정짓는 가장 근원적인 속성을 말한다. 우리가 밥상을 책상의 대용으로 사용할 수 있지만, 이때 밥상을 책상이라고 하지는 않는다. 책상이라는 아이덴티티를 결정짓는 요인은 책을 읽거나 글을 쓰거나 사무를 보거나 할 때에 앞에 놓고 쓰는 상을 의미하며 이러한 속성은 그 위치가 교실에 있는지 공부방에 있는지 변하지 않는다. ‘독립적인 집합’이라는 의미는 어떤 다른 집합과는 본질적으로 다른 자기 혼자만의 고유한 집합이어야 한다는 것이다. 또 집합이 클래스가 되기 위해서는 집합을 구성하고 있는 개별 개체들이 모두 동질성을 가져야 한다. 이는 온톨로지 정의서를 기준으로 해야 한다. 클래스명을 ‘음식’이라고 할 때 정의를 한국에서 요리되는 모든 음식으로 정의한다면 ‘짜장면, 짬뽕, 설렁탕, 비빔밥, 카레’ 등이 모두 포함되나, 한국의 전통음식으로 한정한다면 ‘짜장면, 짬뽕, 카레’ 등은 동질성을 잃어버리는 것이다. 그러나 집합이라고 해서 모두 클래스가 되는 것은 아니다. 예컨대 ‘직원’ 클래스가 회사에서 일하는 모든 사원이나 임원들로 정의한다면, ‘정규직’이나 ‘계약직’은 비록 두 개 이상의 속성과 개체를 갖고 있어도 이를 동일 수준의 클래스로 선언하지 않는다.

溝口理一郎은 인스턴스를 ‘클래스와 인스턴스의 추상 계층에서 최하위층에 존재하는 것으로서 그것보다 하위의 것을 인정할 수 없는 것’이라고 정의하였다.¹⁸⁵⁾ 인스턴스는 온톨로지

182) 류철중·윤보현·김용, 「언어자원 자동 구축을 위한 위키피디아 콘텐츠 활용 방안 연구」, 『디지털융복합연구』, 제13권, 5호, 2015, 229쪽.

183) 김현, 「국립한글박물관 디지털 아카이브 구축 기본 구상」, 한국학중앙연구원 한국학대학원 문화예술학부, 2013, 17~18쪽

184) 한국전산원, 「웹 온톨로지 개발지침 연구」, 2004, 58쪽.

185) 溝口理一郎(저)/최기선·황도삼(역), 『(차세대 웹과 지식 처리의 핵심 기술) 온톨로지 공학』, 두양사, 2009, 128쪽.

설계 목적에 따라 더 이상 관리할 대상이 아니어서 상세화나 구체화가 필요 없는 것을 말한다. 이 말은 하나의 개체를 더 이상 분해할 수 없을 때까지 분해한다는 의미가 아니다. 예를 들어 ‘자동차’라는 클래스를 분류하고 하위 클래스와 인스턴스를 추가해야 할지 결정해야 한다면 이때의 클래스와 인스턴스는 온톨로지를 작성하는 목적에 따라 달라진다. 만약 자동차를 판매하는 영업소에서는 자동차라는 클래스가 최종 노드가 될 것이다. 그러므로 자동차라는 클래스에 들어가는 인스턴스는 그랜저, 마티즈, 스포티지 등의 차종이 될 것이다. 그러나 자동차 정비소라면 자동차가 최종 관리 대상이 아니며 자동차 부품까지 세부적으로 관리해야 할 것이다. 그러므로 하위 클래스를 추가해서 개별 부품별로 클래스를 두고 자동차 부품을 별도로 관리할 수 있을 것이다.

온톨로지가 지식의 소통과 공유를 기본 사상으로 추구한다는 점에서도 온톨로지는 의미적인 관계를 적절하게 표현할 수 있는 방식으로 설계되어야 한다. 클래스의 계층 구조를 설계하는 것은 디렉터리 구조를 만들어 가는 것과 유사하다고 한다. 디렉터리의 계층 구조를 스스로 잘 파악하고 있으면 다른 사람이 보기에 이상해도 큰 문제가 되지 않는다. 그렇지만 클래스의 계층 관계가 논리적으로 잘 설계되지 않으면 의미 관계를 적절하게 표현할 수 없게 되며 온톨로지를 통해서 다른 사람들과 지식을 공유하는 것도 불가능하게 된다.¹⁸⁶⁾

온톨로지 설계와 관련하여 자주 하는 실수 중 하나가 바로 하위 클래스와 인스턴스를 혼동하는 것이다. 예를 들어 ‘項羽’와 신분 중의 하나인 ‘王’이라는 개념 사이의 관계를 살펴보면 ‘王’이라는 범주에 ‘項羽’가 포함되어 있는 것은 분명하다. 그러나 ‘項羽’는 王의 하위 클래스가 아니라 인스턴스로서 파악해야 한다. ‘王’은 집합을 개념화한 클래스이지만 ‘項羽’는 집합의 구성 요소인 인스턴스이기 때문이다.

다음으로 클래스를 순환되는 구조로 설계해서는 안 된다. 아래 그림과 같이 B가 A의 하위 클래스이며, C는 B의 하위 클래스이며, 다시 A가 C의 하위 클래스라면 클래스의 계층 구조에 순환이 있다고 하며, 이 경우에는 A와 B, C가 같다는 것을 선언하는 것이다.

186) 神崎正英(저)/황성역·양혜술(역), 앞의 책(2008), 169쪽.

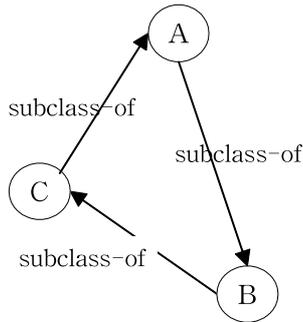


그림 30. 순환 관계의 예

나) 네이밍 규칙

온톨로지 설계에서 클래스와 프로퍼티 등에 일정한 규칙에 의해 일관성 있는 이름을 부여하는 것은 품질면에서도 매우 중요하다. 본 논문에서 온톨로지 자원의 이름을 부여하기 위해 적용한 규칙은 다음과 같다.¹⁸⁷⁾

- OWL, RDF 또는 RDFS 표준에서 기본적으로 제공하는 어휘를 추가로 재정의 하지 않는다.
- 기본적으로 CamelCase 방법을 사용하여 명명한다.(복합어로 된 이름은 첫 글자를 대문자로 하며 단어 간 공백이 없이 쓰이는 명명 스타일)
 - ☞ rdfs:subClassOf, rdfs:subPropertyOf
- 클래스 이름은 대문자로 시작한다.
 - ☞ owl:Class, owl:Restriction
- 프로퍼티 이름은 소문자로 시작한다.(첫 번째 문자를 제외하고 나머지 이름은 CamelCase에 따름)
 - ☞ owl:inverseOf, rdfs:subClassOf
- 클래스 이름은 영문으로 표기하며 단수 명사를 사용한다.
 - ☞ owl:DatatypeProperty, ex:Woman
- 행위를 나타내는 Action은 정확한 의미 전달을 위해 한자 혹은 한글의 기본형으로 표현한다.
- 인스턴스 이름은 한글과 한자를 그대로 사용한다.
 - ☞ ex:項羽, ex:劉邦

187) Dean Allemang·James A. Hendler(저)/김성혁 외(역), 앞의 책(2008), 297쪽.

■ 주어에 대한 부연 설명은 뒤에 information을 추가하는 형태로 명명한다.

☞ bookInformation, personInformation, geoInformation, dateInformation

일정한 원칙에 의해 이름을 부여하는 네이밍이 끝나면, 내용적으로 온톨로지 정의를 가장 잘 표현하는 어휘를 찾아야 한다. 향후 다른 온톨로지와 통합을 염두에 둔다면 더욱 그렇다. 본 연구에서는 적절한 온톨로지 어휘를 선택하기 위해서 Schema.org¹⁸⁸⁾과 CBDB 및 국내에서 발행된 LOD 온톨로지 모델을 참조하였다.

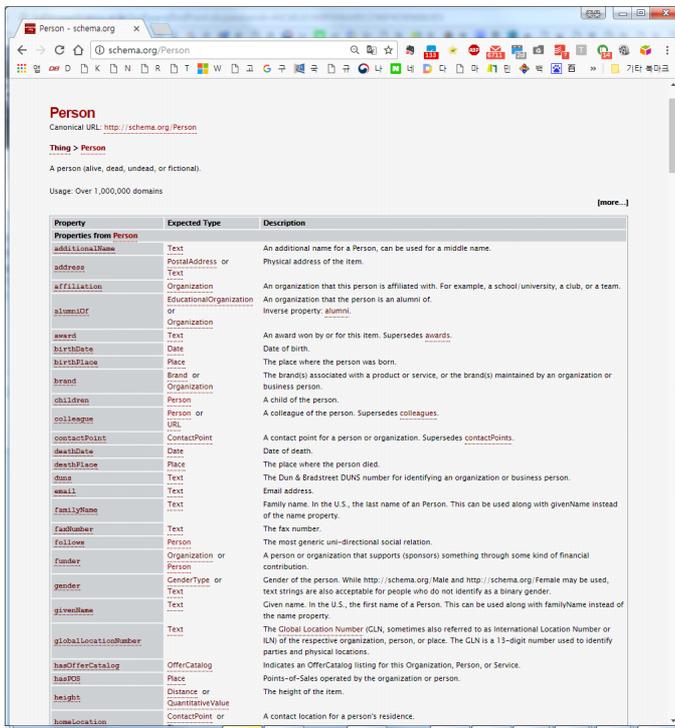


그림 31. Schema.org의 Person에 대한 메타데이터

CBDB(中國歷代人物傳記資料庫, CBDB)는 唐나라부터 淸나라까지 중국의 주요 인물의 개인 전기 자료를 모아서 정리한 데이터베이스로 누구든지 MDB 포맷으로 만들어진 데이터를 무료로 다운받을 수 있다. 인물의 생몰년뿐만 아니라 지역, 관직, 저작 및 인물 관계 정보를 제공하고 있다. 테이블명과 컬럼명을 영어로 했으며, 상당수의 어휘가 영어로 번역되어 있으므로 클래스와 프로퍼티 이름을 영어로 네이밍하는 데 참고가 될 수 있다.

188) Schema.org, <http://schema.org>

2) 개체명 도출

온톨로지를 개발하기 위해서는 먼저 명세화와 개념화 과정을 거쳐야 한다. 명세화는 목적과 질의 내용 등을 구체화하는 것이며, 개념화는 개체가 가지고 있는 공통적인 속성을 추출해서 그것들을 대표할 수 있는 포괄적인 개념을 뽑아 내는 것을 말한다. 그런데 포괄적인 개념을 추출하기 위해서는 대상 자료 내에 어떤 개체가 포함되어 있는지 확인해야 한다. 특히나 본 논문에서 다루는 행위와 개체 간의 관계를 표현하기 위해서는 문서를 직접 확인해 보는 것이 중요하다. 본문 중에 출현하는 개별적인 개체를 온톨로지 관점에서 인스턴스라고 하면 개체명은 Class에 해당한다. 예를 들어 주요 개체명 중 하나인 인명은 온톨로지에서 Person 클래스로 정의할 수 있으며 특정 인물 중의 하나인 ‘項羽’는 인스턴스에 해당한다.

역사서에는 어떤 개체들이 포함되어 있는지 확인하기 전에 일정한 기준을 가지고 접근하는 것이 시행착오를 줄일 수 있으므로 개체명 파악을 위해 개체명의 연구 현황을 살펴 보자. 먼저 한국정보통신기술협회에서 정한 정보통신단체표준(국문표준) 중 「개체명 태그세트 및 태깅 말뭉치(Tag Set and Tagged Corpus for Named Entity Recognition)」에서는 개체명을 아래와 같이 15개로 구분하였다. 학문분야, 시간, 문명, 수량 등은 현대문에 주로 나타나는 개체이며 나머지는 역사서에서도 빈번하게 출현하는 개체이다.¹⁸⁹⁾

189) 한국정보통신기술협회, 「개체명 태그세트 및 태깅 말뭉치(Tag Set and Tagged Corpus for Named Entity Recognition)」, 2015, 13쪽~17쪽.

표 23. 정보통신단체표준 개체명 태그세트 및 태깅 말뭉치

연번	대분류 항목	내용
1	PERSON(PS)	인물
2	STUDY_FIELD(FD)	학문분야
3	THEORY(TR)	이론
4	ARTIFACTS(AF)	인공물
5	ORGANIZATION(OG)	기관
6	LOCATION(LC)	지역
7	DATE(DT)	날짜
8	TIME(TD)	시간
9	CIVILIZATION(CV)	문명
10	ANIMAL(AM)	동물
11	PLANT(PT)	식물
12	QUANTITY(QT)	수량
13	EVENT(EV)	이벤트
14	MATERIAL(MT)	물질
15	TERM(TM)	용어

역사서에서 출현하는 개체명을 파악하기 위해서는 색인을 참조할 필요가 있다. 한국고전번역원에서는 1993년부터 2004년까지 『한국문집총간』에 담긴 주요 용어에 대한 색인집을 간행하는 사업을 수행하여, 조선 전기까지 간행된 문집 212종의 용어 색인을 간행한 바 있다.¹⁹⁰⁾ 색인을 수집할 때 참조한 분류 체계는 아래 표와 같다. 그 뒤 『한국문집총간』 색인집 간행은 이후에 고전용어 시소러스 개발 형태로 이어진다. 2005년 하반기부터 교육인적자원부가 지원하는 국학진흥사업의 일환으로 “고전용어 시소러스 개발 사업”을 추진하게 되었다. 이 사업에서는 색인의 분류를 3단계로 구분하였으며, 그중 1단계에서는 ‘인명, 자연물, 문헌, 유적, 물품, 법제, 시간, 지명, 사상, 사건, 기타’로 분류하였다.

표 24. 『한국문집총간색인』 등재 용어의 유형 분류¹⁹¹⁾

구분	범례
----	----

190) 김현, 『인문정보학의 모색』, 북코리아, 2012, 527쪽.

191) 김현, 「한국 고전적 전산화의 발전 방향」, 『민족문화』, 제28집, 한국고전번역원, 2005, 172~173쪽.

구분		범례
인물	인명	<ul style="list-style-type: none"> · 성명, 별호(자, 아호, 호, 시호, 봉호, 당호) · 왕의 호칭 · 종실 · 의빈의 호칭 · 왕비, 공주,옹주의 호칭 · 기생, 노비, 여자의 호칭 · 혼합 호칭 또는 전성된 호칭 ('성+관직명', '지명+존칭' 등)
공간	지명	<ul style="list-style-type: none"> · 일반지명 · 자연지명 · 관향 · 가상의 지명 · 막연한 지명 (방위 등으로 특정 지역을 일컫는 경우)
	국명	<ul style="list-style-type: none"> · 국명, 왕조명 · 수식어가 붙은 국명
	건물명	<ul style="list-style-type: none"> · 궁전의 명칭 · 樓, 臺, 亭, 館, 院, 書院, 鄉校, 橋 등 · '人名, 地名+廟, 祠' 형태의 건축물명
시간	연호	<ul style="list-style-type: none"> · 왕의 연호
시간	간지, 절기, 월, 일	<ul style="list-style-type: none"> · 연 간지 · 절기명, 월의 이칭, 날의 이칭, 명절 및 날짜로 된 절기
서명	서명, 작품명	<ul style="list-style-type: none"> · 경전, 문집의 본서명과 이칭, 약칭 · 판본명 · 지명 + 錄, 誌 · 중국고전의 편 · 장 · 폐명 · 그림 이름 등의 작품명
용어, 사물	관직명, 관칭명	<ul style="list-style-type: none"> · 관직 관칭의 공식명칭 및 이칭 · 고사에 의거해 쓰인 명칭 · 임시관칭명 · 사신 관계 용어 · 공신호
	용어	<ul style="list-style-type: none"> · 고사용어 · 숫자용어(숫자와 함께 구성된 용어로서 관용적으로 쓰이는 것) · 제도, 세금 관련 용어 · 과거 시험의 명칭 · 제사이름이나 상례 용어 · 성리학 관련 용어 · 도가, 불교 관련 용어 · 日蝕, 月蝕, 地震 등 특별한 자연이나 천문 현상을 나타내는 용어 · 서적 간행 형식에 관련된 용어나 판본 및 간본에 관련된 용어
	사물	<ul style="list-style-type: none"> · 수식어가 붙어 성어를 이룬 사물의 이름 · 특정 사물에 대한 관용적인 이칭 · 고사와 관련된 사물 · 별 이름

다음으로는 위의 자료를 참조하여 실제 역사 자료에 적용했을 때 어느 정도 적용할 수 있을지 확인하기 위해 『通鑑節要』 XML 데이터 내에 주요 개체에 대해 직접 마크업하였다. 아래의 표는 최초로 개체명 태깅을 위해 정의한 항목에, 직접 마크업을 하면서 발견된 항목을 추가하고, 또한 역사서를 표현하는데 필요하다고 판단되는 항목을 추가하여 최종적으로 정리한 목록이다.

표 25. 『譯註 通鑑節要』의 원문과 번역문에 태깅한 개체명 예시

대분류	중분류	내용
인명(person)	人名	사람의 인명
	字	사람의 자
	號	사람의 호
	尊號	존호
	謚號	시호
	廟號	묘호
	年號	연호
	堂號	서실, 서실 이름 등
	法號	법호
	法名	법명
	初名	초명
	初字	초자
	本名	본명
別稱	별호, 사람 이름, 사물의 별칭, 이칭 등	
관직(position)	官職	벼슬이름, 관작
관서(office)	官署	전통 시대의 관서
기관·	機關	기관, 학교 등
단체(organization)	團體	군대, 학과 등
국가명(nation)	國籍	사람 주어가 태어난 국가, 국적
	王朝名	전통 왕조
	時代	사물 주어의 배경이 되는 시대
지명(place)	行政區域	州, 縣 등
	住所	소재지, 땅이름 등
	墓所	묘, 세실, 종묘 등
	自然地名	산, 강, 호수 등
	人文地名	사람의 힘에 의해서 만들어진 지역
건축물(building)	建築物	건축물, 시설물
서명(book)	書名	책, 문집 등
	篇名	편명
작품(works)	作品名	그림, 시, 글귀, 음악, 미술작품, 고전시가명, 곡명, 林華人傳 등의 傳
문서(document)	文書	문서, 서류, 조령, 칙서 등
물명(thing)	服飾	의복, 복식
	器具	도구
	商品	장신구 등 기념물

	軍事	군사, 무기 등
제도(system)	制度	제도, 과거시험 등
법률(law)	法律	법률, 형벌
시간(time)	節氣	24절기
	時間	년, 월, 일, 간지 등
연호(era)	年號	연호
동물(animal)	動物	가축, 어류, 곤충 등 모든 동물
식물(plant)	植物	나무, 풀 등
식품(food)	食品	음식, 과일
신분(status)	身分	황제, 왕 등의 신분
직업(occupation)	職業	직업
천문(astronomy)	天文	별자리, 행성(行星), 항성(恒星), 성단(星團), 성운(星雲), 은하(銀河), 일식, 월식 등
기상(weather)	氣象	날씨, 태풍 등
풍습(custom)	風習	전통, 풍속 등
종교(religion)	宗教	佛敎, 道敎 등
사건(event)	事件	동사, 형용사(행위와 상태)를 나타내는 서술어
용어(term)	貨幣	貨幣
개념(concept)	概念	道, 仁 등의 개념
이론(theory)	理論	
고사성어(idiom)	故事成語	錦衣夜行, 四面楚歌 등의 고사성어
종족(tribe)	宗族	部族, 氏族, 宗族, 民族
상(prize)	賞	
행위(act)	行爲	공격하다, 죽이다, 묻다 등

『通鑑節要』에서 나타난 개체명을 태깅한 결과는 다음과 같다. 실제로 전체 개체명 중에서 인명이 압도적으로 많은 비율을 차지하고 있음을 알 수 있다.

표 26. 『通鑑節要』에 나타난 개체명 분포

연번	요소명	원문	비율	비고
1	person	23,123	50.47%	인명, 자, 호 등 포함
2	place	5,296	11.56%	
3	nation	4,700	10.26%	
4	position	3,643	7.95%	
5	time	2,793	6.10%	
6	idiom	1,962	4.28%	
7	book	1,339	2.92%	
8	era	957	2.09%	
9	term	772	1.69%	
10	tribe	590	1.29%	
11	building	200	0.44%	
12	office	108	0.24%	
13	status	84	0.18%	
14	system	74	0.16%	
15	thing	53	0.12%	
16	animal	45	0.10%	
17	concept	33	0.07%	
18	astrology	35	0.08%	
19	group	8	0.02%	
합계		45,815	100%	

이렇게 마크업된 요소는 지식 관계망을 통해 새로운 지식으로 연결시키는 서비스를 보여 줄 수도 있다. 『通鑑節要』를 단순히 역사의 탐구뿐만 아니라 『通鑑節要』에 나타난 사건 등을 통해서 문화적으로 관련된 또 다른 지식으로 확대해 나갈 수 있는 요소로 활용할 수도 있다. 다음은 지금까지 도출된 개체명을 기본으로 하여 온톨로지 설계를 진행한다.

3. 클래스 설계

가. URI 설계

기계가 웹상의 자원을 이해하기 위해서는 수많은 개념을 자체적으로 구별할 수 있도록 해야 한다. 이러한 구별 수단 중 하나가 URI 참조(URIref)이다. URIref는 절대로 중복될 수

없는 유일성을 가지고 있다. 그러므로 ‘사과(apple)’에 대한 URIref가 ‘http://Ontology.kcisa.kr/ontobook/plant/fruits#apple’이고, ‘사과(apology)’에 대한 URIref가 ‘http://dictionary.kcisa.kr/language/words#apology’라면 컴퓨터는 URIref가 지시하는 개념이 어떻게 정의되어 있는지, 그리고 ‘사과’가 무엇을 의미하는지 정확하게 이해하게 된다. 그런데 앞서 URI 참조에서 URI 주소 간에는 어떠한 의미적인 필연 관계가 없다고 하였다. 즉 ‘Apple’과 ‘Apology’이 어떠한 계층 구조에 의해 정해진 URI 체계가 없다는 것이다. 단지 URI 주소가 서로 다르기만 하면 된다. 실제로 위키데이터에서는 Apple은 Q312으로, Apology은 Q4780433으로 ID가 부여되어 있다.¹⁹²⁾ 문제는 URI 참조는 결국 사람이 부여해야 하고 관리해야 한다는 점에 있다. 결국 사람이 이해하기 쉬운 URI 체계가 필요한 것이다.

URI 주소 체계는 기존 자원의 재활용 측면에서 많은 사람들이 이용하는 URI 참조를 사용하는 것이 좋다. 예를 들어 인명에 대해 URI를 부여할 경우에 기존에 발행되어 있는 위키피디아, DBpedia, WIKIDATA를 활용할 수 있다.

URI 주소 체계를 설계할 때 URI 체계에 의미를 부여하는 방법과 단순 일련번호로 하는 방법이 있다. 이 두 가지 방법 모두 개체를 식별해야 한다는 기본 전제가 깔려 있다. 이에 대해 정한민은 URI는 해당 자원을 유일하게 식별할 수 있는 ID의 일종으로 네임스페이스(Namespace), 접두어(Prefix), 식별자(identifier)로 구성되어 있으며, 네임스페이스는 이름을 구분짓기 위한 추상적 그룹의 집합체이고, Prefix는 식별자 부여 정책 충돌을 해소하기 위한 구분자라고 하였으며, 동명이인 문제로 인해 문자열만으로 사람을 구분해서도 안된다고 하였다.¹⁹³⁾ 윤소영은 URI에 대해 일련번호 코드나 시스템 id 등을 이용하는 방법도 제시되고 있는데 이는 유일성을 보장하나, 규칙성이 없고 데이터 이용자도 불편하게 만드는 URI 생성 방법이라 하였다. 그리고 유일성을 보장할 수 있는 적절한 이름 규칙을 가지도록 하며 데이터를 활용하려는 이용자도 알 수 있도록 직관적인 형태를 유지하도록 할 것을 제시하였다. ¹⁹⁴⁾

한국정보화진흥원에서 간행한 「링크드 데이터 구축 공정 가이드 v1.0」에 따르면 URI를 발행하는 기준을 자원 유형에 따라 별도의 URI로 구분하고 각 URI 유형별로 별도의 URI 규칙을 따르도록 되어 있다.¹⁹⁵⁾ URI를 생성할 때 객체에 대한 정보와 객체 자체를 지칭하는 것을 구분할 수 있도록 해야 하는 것이 바람직하다.¹⁹⁶⁾ 그리고 URI 설계 원칙으로 몇

192) Wikidata, https://www.wikidata.org/wiki/Wikidata:Main_Page

193) 정한민·강인수·구희관·이승우·성원경, 「URI 서버에 기반한 국가 R&D 기반정보 온톨로지 설계 및 구현」, 『정보관리연구』, 제37권, 제2호, 2006, 111쪽.

194) 윤소영, 앞의 논문(제30권 제1호, 2013), 274쪽.

195) 한국정보화진흥원, 「링크드 데이터 구축 공정 가이드 v1.0」, 2014, 67~73쪽.

가지를 제시했는데 그 중에서 중요한 몇 가지를 열거하면 아래와 같다.¹⁹⁷⁾

- HTTP를 활용해서 URI가 해석 가능 하도록 함(resolve)
- 명시적인 URI의 형태를 나타내기 위해 일관된 경로 구조를 사용
- 데이터 발행자(publisher)는 데이터 셋이 다른 분야 혹은 관련된 분야에서 재사용이 가능한지 여부를 확인해야 함
- 최소한 기계가독형 Representation URI를 제공해야함
- URI 구조에는 session ID와 같이 변경될 가능성이 있는 것을 포함시켜서는 안됨
- URI 경로 구조는 사람이 읽을 수 있는 형태로 URI가 대표하고 있는 내용을 합리적으로 이해할 수 있어야 함

또 같은 자료에서 URI 구문을 생성할 때 객체에 대한 정보와 개체 자체를 지칭하는 것을 구분할 수 있도록 해야 한다고 하고 그 예로 DBpedia를 들었다. DBpedia는 아래와 같이 URI 주소를 Resource와 Page로 구분하여 서비스하고 있다.¹⁹⁸⁾

- <http://dbpedia.org/page/Seoul>
- <http://dbpedia.org/resource/Seoul>

정리하면 기계 가독형이면서도 사람 또한 이해할 수 있는 주소 체계를 권고하고 있는 것이다. 위의 연구와 지침을 참조로 하여 본 연구에서 다음과 같은 주소 체계를 설계한다. 먼저 클래스의 분류이다. 본 연구는 역사 분야 전체를 다루다보니 같은 이름의 개체가 상당수 발견된다. 예를 들어 ‘甘露’는 일반적인 물명으로서의 ‘甘露’와 年號로 사용되는 ‘甘露’가 있다. 『표준국어대사전』에 의하면 甘露의 뜻으로 다섯 가지가 제시되어 있다. 이 중 1부터 3까지는 물질을 의미하며, 4는 상상 속의 물질이며 5는 불교 용어이다.

- 「1」천하가 태평할 때에 하늘에서 내린다고 하는 단 이슬.
- 「2」생물에게 이로운 이슬.
- 「3」여름에 단풍나무·떡갈나무 따위의 잎에서 떨어지는 달콤한 액즙. 진드기가 식물 세포 속의 탄수화물, 단백질 따위를 흡수하여 단 즙을 만들어 배설한 것이다.
- 「4」『불』도리천에 있다는 달콤하고 신령스러운 액체. 한 방울만 먹어도 온갖 번뇌와 고통이 사라지며 죽지 않고 오래 살 수 있다고 한다.
- 「5」『불』부처의 가르침이 중생에게 달콤하고 이로운 이슬이 됨을 비유적으로 이르는 말

정확한 의미를 전달하기 위해서는 이들을 의미별로 세밀하게 구분해 주어야 한다. 그렇게 되면 의미의 혼동이 발생하지 않고 정확한 의미 전달이 가능해 진다. 그런데 이러한 일을

196) 한국정보화진흥원, 앞의 자료(2014), 69쪽.

197) 한국정보화진흥원, 앞의 자료(2014), 71쪽.

198) 한국정보화진흥원, 앞의 자료(2014), 71쪽.

어느 한 개인이 하는 것은 사실 상 불가능한 일이다.

본 연구에서 기본 URI는 'http://dh.aks.ac.kr/ontologies/historicevent/kno#'로 하였다. 그러다 보니 물명인 '甘露'와 연호인 '甘露'를 구분할 수 없었다. 이들 어휘를 식별하기 위해 먼저 클래스 이름을 네임스페이스로 구분해 주었다. 예를 들어 年號는 네임스페이스를 'knoEraName'를 사용하여 구분하였으며, 물명은 'knoObject'로 구분해 주었다. 이들은 전체 URI 주소로는 'http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#甘露'과 'http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#甘露'로 표현된다.(실제 OWL 문서는 아래에 나와 있다.) 이로써 클래스 간의 중복 문제는 해결하였다.

```
<!DOCTYPE rdf:RDF [
<ENTITY knoThing "http://dh.aks.ac.kr/ontologies/historicevent/thing/kno#" >
<ENTITY knoEraName "http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#" >
]>

<rdf:RDF
xmlns:knoEraName="http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#"
xmlns:knoThing="http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#">

<!-- http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#甘露 -->
<owl:NamedIndividual rdf:about="&knoEraName;甘露">
  <rdf:type rdf:resource="&kno:EraName"/>
</owl:NamedIndividual>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#甘露 -->
<owl:NamedIndividual rdf:about="&knoObject;甘露">
  <rdf:type rdf:resource="&kno:Object"/>
</owl:NamedIndividual>
```

다음으로는 동일한 클래스 간의 중복을 해결해야 한다. 위에서 연호로 사용된 甘露는 중국 역사상 여러 왕조에서 사용하였다.

- 西漢 孝宣帝의 연호(B.C. 52~B.C. 49)
- 三國時代 魏 高貴鄉公의 연호(256~260)
- 三國時代 吳 末帝의 연호(265~266)
- 五胡十六國時代 前秦의 연호(359~364)
- 오대시대 唐 遼東 丹倍等の 연호(926~952)

이들은 구분하기 위해서 위의 방법처럼 네임스페이스를 이용하여 식별하려면 각 왕조별로 네임스페이스를 구분해 주어야 하기 때문에 활용 측면에서 복잡해진다. 이를 해결하기 위해서 '甘露_西漢' 식별자 뒤에 왕조명을 추가하는 방법으로 해결하였다. 이때 한글 이름은 음가문제가 있으므로 한자로 직접 표현한다.

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#甘露 -->

<owl:NamedIndividual rdf:about="&knoObject;甘露">
  <rdf:type rdf:resource="&kno;Object"/>
</owl:NamedIndividual>
<!-- http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#甘露_前秦 -->

<owl:NamedIndividual rdf:about="&knoEraName;甘露_前秦">
  <rdf:type rdf:resource="&kno;EraName"/>
</owl:NamedIndividual>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#甘露_吳 -->

<owl:NamedIndividual rdf:about="&knoEraName;甘露_吳">
  <rdf:type rdf:resource="&kno;EraName"/>
</owl:NamedIndividual>

```

위의 방법을 근거로 본 연구에서 사용한 네임스페이스는 아래와 같다.

표 27. 본 논문에서 사용한 네임 스페이스

네임스페이스	Full URI
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
skos	http://www.w3.org/2004/02/skos/core#
kno	http://dh.aks.ac.kr/ontologies/historicevent/kno#
dcterms	http://purl.org/dc/terms/
knoPerson	http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#
knoPlace	http://dh.aks.ac.kr/ontologies/historicevent/Place/kno#
knoPosition	http://dh.aks.ac.kr/ontologies/historicevent/Position/kno#
knoEraName	http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#
knoDynasty	http://dh.aks.ac.kr/ontologies/historicevent/Dynasty/kno#
knoBook	http://dh.aks.ac.kr/ontologies/historicevent/Book/kno#
knoObject	http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#

나. 행위의 주체(Agent)

행위의 주체를 ‘Agent’로 분류하고 하위 클래스로 ‘Group, Faction, Family, Person, School, Tribe’로 구분하였다. 그룹은 집단, 단체, 정당 등을 포괄하는 의미로 사용하였다. 당파는 『通鑑節要』에서도 ‘牛李黨爭’과 같은 朋黨에 대한 내용이 나타나지만 조선 시대 역사를 연구하는 데 있어 朋黨의 이해는 반드시 필요하다. 본 논문은 단순히 중국사를 온톨로지적으로 표현하는 것이 아닌 보편적인 역사적 지식을 LOD로 표현하고자 했기 때문에 이러한 黨派를 ‘Faction’ 클래스로 분류하였다.

표 28. Agent 클래스 분류

클래스		내용
Agent		행위의 주체
	Faction	당파, 파벌
	Family	가문
	Group	집단, 단체 등
	Organization	기관, 단체
	Person	인물
	Tribe	씨족, 부족, 종족, 민족

Agent의 하위 클래스인 Person는 행위의 주체로 가장 많이 등장한다. Person과 관련하여 주의해야 할 사항은 우선 사람을 지칭하는 다양한 명칭의 문제와 周主와 같이 사람을 지칭하나 일반적인 개체로 보기 어려운 경우의 문제, 그리고 불특정한 다수의 사람을 지칭하는 어휘의 처리 문제이다. 먼저 인명의 명칭에 대해서 알아보자. 아래 내용은 『通鑑節要』에서 나타나는 後周의 世宗 郭榮(柴榮)을 표현한 다양한 명칭의 예이다. 이 중에는 世祖처럼 잘못 표현된 인명도 있지만 동일한 사람이 여러 가지 다양한 형태로 표현함을 알 수 있다. 이들은 모두 하나의 URI 참조를 이용하여 표현하는 것이 가장 좋다. URI 후에 상세하게 설명하겠지만 대표명칭을 사용하여 표현하였다.

표 29. 『通鑑節要』에 나타나는 後周 世宗의 표현

後周世宗郭榮	郭榮	성명
後周世宗郭榮	世祖	郭榮의 묘호를 잘못 쓴 오류
後周世宗郭榮	世宗	묘호
後周世宗郭榮	世宗(郭榮)	묘호와 성명
後周世宗郭榮	睿武孝文皇帝	시호
後周世宗郭榮	周主	왕조명과 주의 결합
後周世宗郭榮	晉王	봉작호

『通鑑節要』에는 周主처럼 왕조명에 主자를 붙여서 황제를 표현하기도 한다. 이러한 표현은 『資治通鑑』에서 비롯된 것이다. 司馬光은 정통을 크게 중시치 않았다. 원래 정통이란 『春秋』에서 시작되었다. 혈연적으로는 嫡庶, 사상적으로는 正統과 異端, 분열할거로는 正統과 閏統을 구분하여 왔다. 그런데 북송 중기에 안으로는 당말, 오대의 혼란의 해독이 청산되지 못하였고, 밖으로는 契丹과 西夏의 침입으로 인해 성리학이 추구하는 의리론과 결합하여 正(閏)統論, 無統論, 明統論이 제기되었다. 司馬光도 이러한 시대적 영향을 받아 魏晉南北朝 시대 중에 東晉 南朝의 황제를 ‘帝’, 五胡 北朝의 황제는 왕조 이름에 ‘主’을 붙여 ‘某主’라고 하였다.¹⁹⁹⁾ 이러한 영향은 『通鑑節要』에도 그대로 반영 되었다.

王朝名이나 廟號도 중복되는 것이 적지 않은데 ‘某主’ 형태로 황제를 표시한다면 어떤 황제인지 파악하는 것은 더더욱 쉽지 않다. 예컨대 본문 중에 ‘梁主’로 표현되어 있는 인물은 당시 년도를 파악하지 않으면 구체적으로 누구인지 알 수 없다. 왜냐하면 중국에 梁나라가 여러 개 나올 수 있으며 한 왕조에도 여러 왕이 있을 수 있기 때문이다. 이들에 대해서 당시 시대와 관련 인물 등을 참고로 하여 ‘梁主’가 구체적으로 누구인지를 정확하게 파악해 주어야 한다. 그래야만 이들 간의 관계를 정확하게 엮어줄 수 있기 때문이다.

199) 이계명, 『자치통감』, 『역사학연구(구 전남사학)』, 제12권, 1998, 138쪽.

표 30. 『譯註 通鑑節要』의 본문 구조 예시

원문	<p><time>四月</time>에 <nation>梁</nation>이 詔以宗廟用牲牢는 有累冥道하니 宜皆以麩爲之하라하니 於是에 朝野諠譁하여 以宗廟去牲은 乃是不復血食이라호되 <person type="" pid="" name="南梁武帝" ref="">梁主</person>竟不從하다</p>
번역문	<p><time>4월</time>에 <person type="" pid="" name="南梁武帝" ref="">梁主</person>가 명령하여 宗廟의 제사에 희생을 사용하는 것은 冥道에 방해가 되니, 마땅히 모두 밀가루로 희생 모양을 빚어 대신하라고 하였다. 이에 朝野가 의론이 분분하여 宗廟의 제사에 희생을 사용하지 않는 것은 바로 다시는 血食을 하지 못하게 한 것이라고 하였으나 <person type="" pid="" name="南梁武帝" ref="">梁主</person>는 끝내 이 말을 따르지 않았다.</p>

원문에서 첫 번째는 ‘梁’은 梁나라인데 번역문에서는 ‘梁主’로 하였다. 여기서 ‘梁主’는 佛敎를 신봉하여 佛敎에 귀의하고 사원을 대대적으로 건축한 南朝 ‘梁武帝’를 지칭한다. LOD를 이용한 역사 자료의 통합을 위해서는 위와 같은 명확한 개체 이외에도 사람이나 개체를 가리키는 것을 모두 마크업해 주어야 한다. 즉 마크업의 범위를 지칭어나 대명사까지도 확대해야 한다. 인명 이외에 사람을 지칭하는 것은 上, 帝, 王, 皇帝, 天子, 皇后 등이 있으며, 사람을 지칭하는 지시어나 官職名, 地名으로도 인물을 나타내는 경우가 적지 않다. RDF 트리를 생성하기 위해선 이들 용어에 대해서 모두 마크업을 해야 한다.

다음으로 불특정 다수의 사람들이 주어로서 역할을 하는 경우이다. 아래 문장에서 주어(agent)는 ‘周兵’과 ‘北漢兵’인데 이들을 URI 참조로 표현해야 할지, 리터럴로 처리해야 할지에 대해서 고민해야 한다. ‘周兵’을 인스턴스로 처리한다면 먼저 ‘周’나라를 구체적으로 식별하여 다른 周나라와 의미의 혼동이 발생하지 않도록 해야 한다. 즉 文王과 武王이 다스린 周나라가 있으며, 이 周나라를 다시 구분하여 西周와 東周로 나누기도 한다. 또한 남북조 시대의 北朝 중 宇文泰의 北周가 있으며, 오대십국 시대의 郭威가 건국한 後周가 있다. 이들을 모두 식별하여 각각에 고유한 식별자를 부여해야 한다.

■ 周兵爭奮하니 北漢兵이 大敗라

☞ 周나라 군사들이 다투어 분발하니, 北漢軍이 크게 패하였다.

또한 인명을 비롯해서 클래스 설계에 있어 고려해야 할 것 중 하나가 자원 이름의 한글과 한자의 표현 문제이다. 나아가 다국어의 표현을 어떻게 해야 할지에 대해서 고민해야 한다. 한글과 한자 문제는 다시 Object Property와 Datatype Property에 따라 달라질 수 있기

때문이다. 이와 관련하여 이전에 발행한 LOD를 현황을 살펴 본다. 아래는 국립중앙도서관에서 ttl 형식으로 제공하는 고서 서지 정보 LOD(nlk_oldbook.tgz)를 RDF/XML 방식으로 변환한 것이다.²⁰⁰⁾ 『通鑑節要』(dcterms:identifier:KOL000009567)는 <http://lod.nl.go.kr/resource/CNTS-00047974125>에 대해서 RDFS에서 제공하는 내장형 프로퍼티 rdfs:label을 이용하여 사람들이 보다 이해하기 쉬운 형태로 표현하였다.

```

<rdf:Description rdf:about="http://lod.nl.go.kr/resource/CNTS-00047974125">
  <rdf:type rdf:resource="http://lod.nl.go.kr/ontology/OldBook"/>
  <rdf:type rdf:resource="http://lod.nl.go.kr/ontology/OnlineMaterial"/>
  <rdfs:label>少微通鑑節要</rdfs:label>
  <extent xmlns="http://bibframe.org/vocab/">13冊(零本): 四周單邊 半郭 23.4 x 16.9 cm, 有界,
10行17字 註雙行, 內向2葉花紋魚尾; 33.4 x 22.2 cm</extent>
  <antecedentSource>원시자료로부터 재생산된 파일</antecedentSource>
  <datePublished rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2015-08-17T1
9:02:41</datePublished>
  <genre>고서</genre>
  <keyword>통감</keyword>
  <keyword>소미통감절요</keyword>
  ..... 생략 .....
  <dc:subject>통감</dc:subject>
  <dc:subject>소미통감절요</dc:subject>
  <dcterms:accessRights rdf:resource="http://lod.nl.go.kr/resource/license0"/>
  <dcterms:hasFormat rdf:resource="http://lod.nl.go.kr/resource/KOL000009567"/>
  <dcterms:identifier>KOL000009567</dcterms:identifier>
  <dcterms:title>少微通鑑節要</dcterms:title>
  <bibo:edition>木板本</bibo:edition>
  <bibo:volume>1-13/ 江贊選</bibo:volume>
</rdf:Description>

```

아래는 국사편찬위원회에서 제공하는 LOD인 nikh.rdf 중에서 忠武公 李舜臣 부분만을 재구성한 것이다. 忠武公 李舜臣에 대해서 URI 참조 'http://lod.koreanhistory.or.kr/인명_이순신'으로 유일성을 확보하였으며, 李舜臣의 이름은 'hlod:realName' 프로퍼티를 이용하여 대표명칭을 표현하고, 'hlod:titleKor' 프로퍼티로 한글 이름을, 'hlod:titleChi' 프로퍼티로 한자 이름을 표현하였다. 字는 'hlod:jaName' 프로퍼티를 이용하여 표현하였으며, 여해(汝諧)처럼 한자를 병기하였다. 이는 한 사람의 字나 號 등은 한 개 이상 있는 사람도 적지 않으므로

200) 데이터 다운로드, 국립중앙도서관, <http://lod.nl.go.kr/home/about/datadownload.jsp>

한글과 한자를 분리하면 이들 간의 연관성을 표현하기 어려워서 그런 것으로 보인다. 이름은 모두 XML에서 스키마에서 제공하는 ‘string’ 방식으로 표현하였다. 다시 정리하면, URI 참조를 이용하여 식별되 그 형식은 ‘인명_일련번호’ 형태로 하였다. 모든 RDF 트리플을 대표하는 명칭은 ‘hlod:realName’이며, 이를 한글과 한자로 분리하여 표현하였다.

```

<RDF:Description RDF:about="http://lod.koreanhistory.or.kr/인명_이순신">
  <hlod:jaName RDF:datatype="http://www.w3.org/2001/XMLSchema#string">여해(汝諧)</hlod:jaName>
  <hlod:posthumousName RDF:datatype="http://www.w3.org/2001/XMLSchema#string">충무(忠武)</hlod:posthumousName>
  <hlod:bonghoName RDF:datatype="http://www.w3.org/2001/XMLSchema#string">덕풍부원군(德豊府院君)</hlod:bonghoName>
  <foaf:gender RDF:datatype="http://www.w3.org/2001/XMLSchema#string">남</foaf:gender>
  <hlod:birthDate RDF:datatype="http://www.w3.org/2001/XMLSchema#string">15450308</hlod:birthDate>
  <hlod:deathDate RDF:datatype="http://www.w3.org/2001/XMLSchema#string">15981119</hlod:deathDate>
  <hlod:birthPlace RDF:datatype="http://www.w3.org/2001/XMLSchema#string">한성(漢城)</hlod:birthPlace>
  <hlod:livingPlace RDF:datatype="http://www.w3.org/2001/XMLSchema#string">아산(牙山)</hlod:livingPlace>
  <hlod:realName RDF:datatype="http://www.w3.org/2001/XMLSchema#string">이순신</hlod:realName>
  <hlod:titleKor RDF:datatype="http://www.w3.org/2001/XMLSchema#string">이순신</hlod:titleKor>
  <hlod:titleChi RDF:datatype="http://www.w3.org/2001/XMLSchema#string">李舜臣</hlod:titleChi>
  <hlod:hasChild RDF:datatype="http://www.w3.org/2001/XMLSchema#string">이회(李薈), 이열(李莳), 이면(李勉), 이훈(李薰), 이서(李莖)</hlod:hasChild>
  <hlod:hasFather RDF:datatype="http://www.w3.org/2001/XMLSchema#string">이정(李貞)</hlod:hasFather>
  <hlod:hasMother RDF:datatype="http://www.w3.org/2001/XMLSchema#string">초계(草溪) 변씨(卞氏)</hlod:hasMother>
  <hlod:hasWife RDF:datatype="http://www.w3.org/2001/XMLSchema#string">상주(尙州) 방씨(房氏)</hlod:hasWife>
  <hlod:hasConcubine RDF:datatype="http://www.w3.org/2001/XMLSchema#string">해주(海州) 오씨(吳氏)</hlod:hasConcubine>
  <RDF:type RDF:resource="http://xmlns.com/foaf/0.1#Person"/>
</RDF:Description>

```

다른 형식을 살펴보면 서소리는 「문화유산 지식 정보 데이터 모델 연구」에서 이 문제에 SKOS의 skos:prefLabel, skos:altLabel을 사용하여 대표명칭과 이칭, 외국어 명칭을 표현하도록 하였다.²⁰¹⁾

201) 서소리, 「문화유산 지식 정보 데이터 모델 연구」, 한국학중앙연구원 석사학위논문, 2014, 82쪽.

```

<bhdm:Title rdf:about="http://digerati.aks.ac.kr/terms/bhdm/Title/불국사삼층석탑명칭">
<skos:prefLabel xml:lang="ko">경주 불국사 삼층석탑</skos:prefLabel>
<skos:altLabel xml:lang="kore">慶州 佛國寺 三層石塔</skos:altLabel>
<skos:altLabel xml:lang="zh-cn">佛國寺三層石塔</skos:altLabel>
<skos:altLabel xml:lang="en">Three-story Stone Pagoda of Bulguksa Temple, Gyeongju
</skos:altLabel>
<skos:altLabel xml:lang="ko">석가여래상주설법탑</skos:altLabel>
<skos:altLabel xml:lang="kore">釋迦如來常住設法塔</skos:altLabel>
<skos:altLabel xml:lang="ko">석가탑</skos:altLabel>
<skos:altLabel xml:lang="kore">釋迦塔</skos:altLabel>
<skos:altLabel xml:lang="ko">무영탑</skos:altLabel>
<skos:altLabel xml:lang="kore">無影塔</skos:altLabel>
</bhdm:Title>

```

지금까지 살펴본 이름을 비롯한 모든 인스턴스에서 고려해야 하는 점은 한글 혹은 한자 등의 다국어 표현 문제, 출현하는 인스턴스의 출현 횟수, 데이터의 형식에 대한 설정 등이었다.

다. 행위(Action)

때로는 ‘행위’와 ‘동작’을 엄밀하게 구분하기도 하지만 본 연구에서는 행위와 동작을 포괄하는 개념으로 사용하였다. 한문에서 행위나 상태를 나타내는 동사는 결국 우리말의 인식 체계에서 이해되어야 한다. 그렇기 때문에 한문에서 사용되는 동사와 별개로 우리말의 동사 분류에 대해서 이해할 필요가 있다. 이숙의는 ‘행위’를 품행이나 실행과 관계되는 ‘움직임, 실시, 수행, 거행, 범죄, 소동, 선행, 악행’ 등의 행동으로 보았으며 동작은 주로 행동주(agent) 자신의 육체적 활동과 관련된 동사들을 ‘동작’으로 구별하였다.²⁰²⁾

본 연구에서는 기본적으로 역사서에 나타나는 수많은 행위를 event 중심으로 하여 意味 役으로 연결하는 것으로 설계하였다. 이때 설계의 기본 방향을 행위는 Action 클래스와 action Object Property로 연결하는 것으로 설계하였다. 이 말은 개체와 개체의 관계를, 프로퍼티에 의해서 표현하던 것을 클래스로 표현한다는 것을 의미한다. 예를 들어 보통 ‘공자가 춘추를 짓다’에서 ‘짓다’라는 동사를 프로퍼티로 표현하는데 본 논문에서는 클래스의 인스턴스로 설계했다는 것을 의미한다. 개체와 개체의 관계를 인스턴스로 표현한다는 것은 새로운 관계가 출현하더라도 설계의 변경 없이 수용할 수 있다는 것을 의미한다. 그렇다면 여기서

202) 이숙의, 「한국어 동사 의미망 구축 연구」, 충남대학교 대학원 박사학위논문, 2006, 63쪽.

중요한 것은 행위나 상태를 표현할 수 있는 분류 체계가 있어야 한다는 것이며 이들 분류 체계는 결국 클래스에 의해 표현된다. 국내 온톨로지는 2000년대 초반까지만 하더라도 대부분 특정 분야의 문서처리를 위한 전문용어 온톨로지에 관한 것이 대부분이었으며, 일반적으로 명사를 대상으로 연구되어 왔다.²⁰³⁾

1) 행위의 표현

앞서 본 연구의 목적은 역사적 지식 추론이 가능한 RDF 트리플로 구현하는 것이라 하였다. 역사적 지식에 나타나는 관계는 정적인 관계와 동적인 관계로 구분할 수 있다. RDF 트리플은 앞서 설명한 것처럼 지식을 컴퓨터가 쉽게 처리할 수 있도록 자원을 주어(Subject)-술어(Predicate)-목적어(Object)로 표현한 논리적인 모델이다. 이러한 모델은 도서관의 메타데이터나 서지 정보 등을 표현하는 데는 유용하나 일반적인 문장을 표현하는 데는 많은 제약이 있다. 예컨대 RDF 트리플로 ‘項羽가 義帝를 죽이다’와 같은 간단한 문장은 표현할 수 있다. 그러나 만약에 여기에 시간과 공간을 추가하려면 쉽지 않다. 아래 그림과 같은 그래프 구조를 근원적으로 허용하지 않기 때문이다. 즉 행위 자체가 발생한 시간과 공간은 단순한 이항 관계로는 표현이 불가능하다.

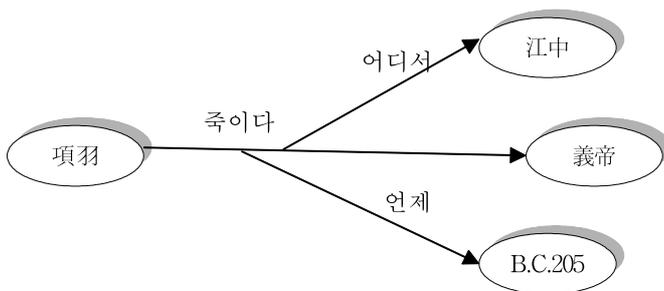


그림 32. 행위에 시간과 공간의 추가된 형태

이를 해결하기 위해서 공백 노드(Blank node)를 사용해서 표현하는 방법과 중계자를 이용해서 표현하는 방법 등이 연구되었다. 공백 노드는 이름을 갖지 않는 노드를 말하며 내부적으로 노드와 노드 사이에 공백 노드를 배치하여 정보를 좀 더 확장적으로 표현하는 방법이다. 예를 들어 아래 그림과 같이 ‘죽이다’라는 속성과 ‘義帝’ 사이에 공백 노드를 두고 그 노드에 장소와 발생 시간을 연결하는 것이다.

203) 이숙의, 『한국어 동사 온톨로지 구축 연구』, 역락, 2013, 44쪽.

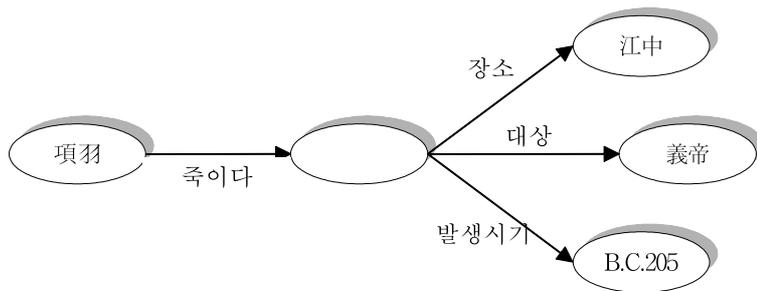


그림 33. 공백 노드를 이용한 시간과 공간의 표현

중개자(mediator)를 이용하는 방법은 중개자 역할을 하는 클래스를 이용하여 한 개체에 관계하는 다양한 관계와 개체를 서로 연결시키는 방법으로²⁰⁴⁾, 공백 노드에 구체적인 이름을 부여했다는 차이가 있을 뿐이다. 그러나 이들 방법은 어느 정도 부가 정보를 표현할 수는 있으나 역사서에 나오는 다양한 사실을 표현하기에는 많은 문제가 있다. 제일 큰 문제는 역사서에 나타나는 관계의 수가 너무 많다는 것이다. 만약 역사서에 나타나는 행위를 프로퍼티를 이용하여 관계를 설정한다면 그 수는 매우 많아 질 것이다. 지금까지 대부분의 온톨로지 설계는 사전에 정해진 관계만을 수용할 수 있었다. 예를 들어 CIDOC-CRM에서는 개체를 E1부터 E94까지 89개를 설정한 후 이들 간의 관계를 P1부터 P168까지 149개를 정해 놓고 'E1 CRM Entity is identified by(identifies) E41 Appellation'처럼 문화유산과 관련된 정보를 온톨로지로 표현할 수 있게 하였다.²⁰⁵⁾ 이들 관계는 미리 정해 놓은 관계가 아니면 새로운 관계가 발견될 때마다 그에 맞는 관계를 추가해야 하며 실제로 버전이 바뀔 때마다 새로운 관계의 유형이 추가되는 것을 확인할 수 있다. 만약 이런 방식을 사용하여 역사서의 본문을 RDF로 표현한다면 어쩌면 국어사전에 등재된 모든 어휘를 프로퍼티로 등록해야 할지도 모른다. 역사서의 특성상 한자어가 많다는 점을 생각하면 국어사전보다 더 많은 관계를 설정해야 할 것이다.

또한 현재까지 공백 노드를 활용하여 문장을 일관성 있게 표현할 수 있는 연구가 되어 있지 않다. 역사서에 담긴 복잡다단한 행위 중에 어떤 것을 선택해야 할지 일정한 기준이 필요하다. 이러한 기준이 마련되지 않으면 역사서의 본문 내용을 LOD로 통합하는 건 애초

204) 김수경, 안기홍, 최호진, 「새로운 N-ary 관계 디자인 기반의 온톨로지 모델을 이용한 문장의미결정」, 『정보관리학회지』, 제25권, 제4호, 2008, 53쪽.

205) cidoc_crm_version_6.2.2 (WorkingDoc).pdf에 의하면 현재까지 개체와 개체의 관계가 149개로 확인되었다.

부터 불가능하기 때문이다.

이러한 문제를 해결하기 위해 본 논문에서는 모든 행위를 이벤트(event)로 보고 이들 행위와 다른 개체와의 관계를 통사론에서 사용하는 意味役²⁰⁶을 도입하여 표현하고자 하였다. 즉 아래 그래프처럼 event라는 기본 노드를 중심으로 문장에 나타는 論項의 역할에 따라 각각 의미역으로 연결을 시키는 방법이다. 이때 event를 위에서 언급한 것처럼 중개자를 사용하여 ‘項羽가 義帝를 죽임’처럼 부여할 수 있으나, 행위가 연속으로 나타날 경우에 event 이름이 복잡해 진다는 현실적인 어려움으로 인해 단순히 일련번호를 부여하는 것으로 하였다.

이러한 의미역으로 관계를 기술할 경우 역사서에 나타나는 행위의 대부분을 큰 제약 없이 표현할 수 있을 것으로 기대된다. 만약에 좀 더 상세한 부사어 등을 표현하고 싶을 경우에는 해당 용어를 포괄할 수 있는 적절한 용어를 사용하여 관계를 추가하면 된다.

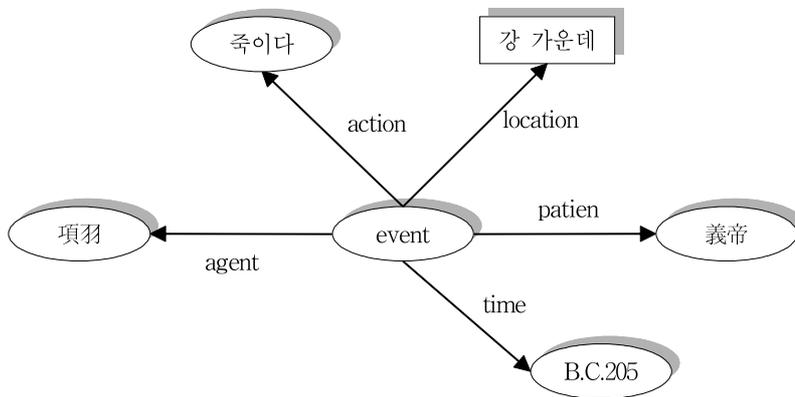


그림 34. 의미역을 이용한 다항 관계의 표현

2) 『通鑑節要』에 나타난 동사의 유형

역사적 사실을 표현하기 위해 이벤트 클래스를 중심으로 각 개체와 의미역을 이용하여 관계를 설정하는 방법을 적용하였다. 이때 행위는 Action 클래스로 분류하며, Action은 action Object Property로 연결한다. Action 클래스로 분류한다는 것은 실제 행위를 인스턴스로 표현한다는 것을 의미한다. 그렇다면 『通鑑節要』에 존재하는 Action 인스턴스를 구분해 내야 한다. 그 방법으로, 전체를 대상으로 본문 중에 중요하다고 생각되는 동사 혹은 행위 명사에 수작업으로 일일이 태깅을 하였다. 그 결과로 『通鑑節要』 48권부터 50권 사이에 본

206) 여기서 意味役은 한 문장이 완성되기 위해 꼭 필요한 문장 성분과 그 문장 안에 있는 동사나 형용사 등의 술어가 맺고 있는 의미적 관계를 말한다.

문 중에 행위와 상태를 나타내는 동사는 모두 1,787개이었으며, 그 중에서 빈도순으로 120개를 제시하면 다음과 같다.

표 31. 행위를 나타내는 동사의 빈도

연번	어휘	한글어휘	빈도	연번	어휘	한글어휘	빈도	연번	어휘	한글어휘	빈도
1	至	이르다01	39	41	出	나오다	7	81	克	점령하다	3
2	遣	보내다	36	42	改	고치다01	6	82	軍	주둔하다	3
3	爲	임명하다01	32	43	亡	망하다	6	83	謀	죽이다01	3
4	卽	즉위하다	30	44	還	돌아오다	6	84	守	지키다01	3
5	稱	칭하다	29	45	爲	세우다01	6	85	薨	죽다01	3
6	爲	삼다02	29	46	上	올리다01	6	86	誅	죽이다01	3
7	爲	되다01	19	47	戰	싸우다	6	87	罷	칭하다	3
8	將	거느리다	18	48	弑	시해하다02	6	88	陳	진열하다01	3
9	殺	죽이다01	18	49	引	이끌다	6	89	作	짓다01	3
10	斬	베다02	16	50	爲	하다01	6	90	總	총괄하다	3
11	卒	죽다01	15	51	賜	하사하다01	6	91	廢	폐하다	3
12	討	토벌하다01	13	52	請	청하다	6	92	曰	하다01	3
13	立	세우다01	13	53	寇	침략하다01	5	93	陷	함락하다	3
14	更	고치다01	13	54	圍	포위하다01	5	94	平	평정하다02	3
15	命	명하다02	13	55	除	없어지다	5	95	敗	패퇴시키다	3
16	入	들어가다01	12	56	破	대파하다01	5	96	平	평정되다01	3
17	降	항복하다	12	57	歸	돌아가다	5	97	寇	침입하다01	3
18	殂	죽다01	12	58	滅	멸망시키다	5	98	反	배반하다01	3
19	立	즉위하다	11	59	詔	명하다02	5	99	謁	배알하다	3
20	攻	공격하다	11	60	破	격파하다	5	100	獻	바치다01	3
21	帥	거느리다	11	61	生	낳다01	5	101	附	붙다	3
22	入	들어오다	10	62	下	내리다01	5	102	襲	세습하다	3
23	進	진진하다02	10	63	出	나가다	4	103	建	세우다01	3
24	據	점거하다02	10	64	驕	교만하다	4	104	有	소유하다01	3
25	發	출발하다	10	65	走	도망하다01	4	105	舉	일으키다	3
26	死	죽다01	9	66	盡	없어지다	4	106	起	일으키다	3
27	懼	두려워하다	9	67	得	얻다01	4	107	進	승진시키다	3
28	擊	공격하다	9	68	疑	의심하다	4	108	勸	오르다	3
29	改元	개원하다01	8	69	作	일으키다	4	109	敗	대패하다	3
30	從	따르다01	8	70	崩	승하하다	4	110	謀	도모하다	3
31	曰	말하다	8	71	禪	선양하다03	4	111	聞	듣다01	3
32	迎	맞이하다	8	72	說	설득하다	4	112	誅	모의하다03	3
33	表	올리다01	8	73	封	봉해지다	4	113	委	맡기다	3
34	封	봉하다02	8	74	焚	불태우다	4	114	餌	먹다02	3
35	奉	받들다	8	75	召	부르다01	4	115	恐	두려워하다	3
36	擒	사로잡다	8	76	不許	허락하지_않다	4	116	進	권하다	3
37	卽位	즉위하다	8	77	許	허락하다	4	117	赦	내리다01	3
38	追	추격하다01	8	78	幸	칭하다	4	118	掠	노략질하다	3
39	逼	핍박하다	7	79	朝	조회하다01	4	119	以	거느리다	3
40	敗	패하다	7	80	取	점령하다	4	120	居	거주하다	3

3) 행위의 대역어의 처리

한문 원문을 우리말로 번역할 때 명사는 대부분 그대로 번역이 되는 데 비해 동사는 상당수 우리말로 풀어서 번역하는 경우가 많다. 『통감절요』는 1,000년이 넘는 중국의 역사를 다루고 있으며, 조선 시대 사대부들의 중국사 인식의 기본 토대를 제공하던 자료라는 점에서 참조할 만한 가치가 있다. 원문 중에 사용된 한자어가 번역문에서 어떻게 번역되었는지 대역어가 잘 정리되면 앞으로 자동 번역 등에도 활용할 만한 가치가 있을 것이다. 『통감절요』에 한자어와 번역된 구체적인 사례는 양방향으로 ‘translationWord’ 오브젝트 프로퍼티로 표현한다. 이때 우리말 어휘는 『표준국어대사전』의 구분 번호로 식별하고, 한자어는 별도의 식별자 없이 텍스트로 식별한다.

클래스		내용
Action	Chinese	한문 동사
	Korean	한글 동사

라. 지명(Place)

지명이란 지표면에 존재하는 지리적 실체(geographical features)의 이름을 말한다. 각각의 지리적 실체들을 가리키는 지명은 인간에 의해 의미와 가치가 부여된 곳을 다른 곳과 구별해 주는 고유명사이다.²⁰⁷⁾ 그런데 지명은 한 번 명명되면 쉽게 변하지 않는 존속성을 가지고 있으면서도 동시에 끊임없이 변화해 가는 가변적인 존재이기도 하다. 지명은 왕조에 따라 그 영역과 명칭이 달라지는 것이 흔한 일이었다. 지명은 인명과 마찬가지로 동일한 명칭이 적지 않게 나타난다. 예를 들어 하버드 대학에서 제공하는 China Historical GIS²⁰⁸⁾에서 ‘江陵’을 검색해 보면 71개의 ‘江陵’이 나타난다.

지명과 관련된 논의는 첫째, 지명의 이칭을 어떻게 처리할 것인가? 둘째, 지명 간의 계층 관계를 어떻게 처리해야 할지, 셋째, ‘河南道宋州碭山縣午溝里’와 같은 행정 구역을 어떻게 표현할지에 대한 것이다.

전통 시대의 행정 구역은 왕조마다 차이가 있기 때문에 그 모든 것을 식별하고, 구분해 준다는 것은 본 연구의 범위를 벗어난다. 이러한 한계는 그대로 인정하고, 구체적인 영역을

207) 국토지리정보원, 「지명 표준화 편람」, 국토지리정보원, 2012, 11쪽.

208) China Historical GIS, 하버드대학교, <http://www.fas.harvard.edu/~chgis>

차치하고 있으며, 동일한 지역을 지칭하면서 명칭이 다른 경우에만 통일화 작업을 하였다. 지명의 문제는 시맨틱 웹의 설계 5원칙으로 설명할 수 있다. 5원칙 중의 하나인 부분적인 정보(Partial information)는, 시맨틱 웹은 완전한 정보를 전제로 하는 것이 아니라 부분적인 정보만 얻을 수밖에 없는 실세계를 대상으로 한다고 하고 있다. 정보는 언제 누가 어디에 기술해도 상관없으며, 같은 자원에 관한 모순된 정보가 존재할 수도 있다. 이렇게 불완전한 정보를 전제로 하고 있으며 이를 통해 유익할 결론을 도출해 내는 것을 목표로 하고 있다.²⁰⁹⁾ 비록 지명의 구체적인 영역을 알 수도 없고, 서로 같은 명칭임에도 시대에 따라 다른 영역을 지칭하는 경우가 있더라도 문제가 되지 않는다는 의미이다.

다음으로 지명 간의 계층 관계는 뒤에서 언급하는 공간적 부분 관계(spatiallyInclude와 spatiallyInside)를 활용하여 표현하였다.

- Dynasty rdf:type 唐
- 唐 spatiallyInclude 京畿道
- 京畿道 spatiallyInclude 虢州
- 虢州 spatiallyInside 京畿道

다음은 행정 구역 명칭의 문제인데, 後梁 太祖 朱全忠은 ‘河南道宋州礪山縣午溝里’²¹⁰⁾에서 태어났는데 이들 ‘河南道, 宋州, 礪山縣, 午溝里’ 등은 모두 하나의 지명으로서의 자격을 갖추고 있으면서도 이들 간에 상하 관계 혹은 포함 관계 등이 형성되어 있다. 이런 행정 구역 명칭은 각각을 단위를 하나로 모아서 표현해도 되고, 다항 관계로 표현할 수도 있다. 그런데 역사서에서 이런 전체 행정 구역이름이 그대로 나오는 경우는 그리 많지 않고, 이 중에서 이름 이름이 나타난다. 이들은 전체 명칭을 인스턴스로 등록할 수도 있다. 그러나 역사서에 나타나는 행정 지명을 온전하게 전체 이름을 적어주는 경우가 그리 많지 않다. 전체 지명을 리터럴로 처리할 수 있다. 본 연구에서는 행정 구역별로 클래스를 분류하고 공백 노드를 이용하여 행정 구역별로 구분해 줄 것을 제안하다.

209) 神崎正英(저)/황성역·양해술(역), 앞의 책(2008), 5쪽.

210) 『舊五代史』一卷, 梁書一, 太祖紀第一에 의하면 朱全忠은 礪山縣午溝里에서 태어난 것으로 되어 있다.

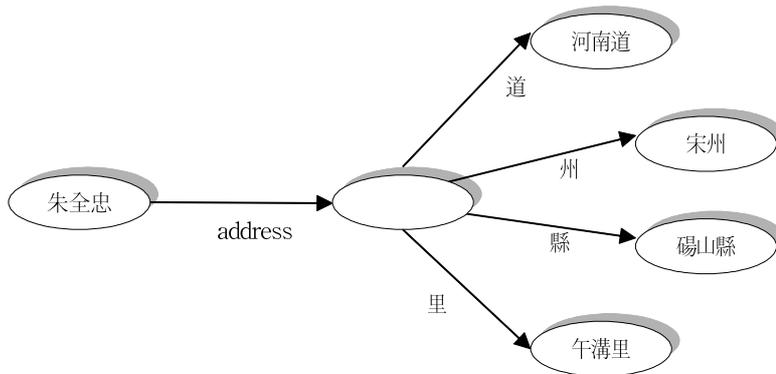


그림 35. 행정 구역의 표현 방법

마. 왕조(Dynasty)

역사에서 왕조는 일반적인 지명과 다른 특징을 지니고 있다. 왕조와 지명이 모두 일정한 공간을 기반으로 하고 있다는 점은 유사하나, 지명은 구체적인 시점과 종점이 존재하지 않는 반면에 왕조는 시점과 종점이 명확하게 존재한다. 그리고 연호 등의 정보와도 연계시켜야 할 부분이 많기 때문에 지명과 분류해서 관리하는 것이 더 효율적이다. 이 때문에 왕조를 지명에서 분리하여 별개의 ‘Dynasty’이라는 클래스로 분류하였다.

중국사에서 『漢書』, 『後漢書』 등 대부분의 역사서가 王朝를 중심으로 편찬되었다. 그런데 역사서에는 같은 이름을 가진 王朝가 적지 않게 등장한다. 이런 혼란 때문에 역사가는 史書를 편찬할 때 각 왕조를 구분하려고 지역에 따라 東·西를 붙이거나 시기에 따라 前·後를 붙이기도 하고, 姓氏를 앞에 붙이기도 하였다.²¹¹⁾ 온톨로지를 구축할 때도 마찬가지로 모든 왕조는 예외 없이 유일성을 확보해야 한다. 예를 들어 중국에서 ‘梁’을 국호로 삼은 왕조를 위키피디아에서 조사하면 아래와 같이 11개나 된다. 이러한 王朝는 역사서를 읽는 중에 많은 혼란을 주며 연계 및 통합에도 문제를 일으킨다. 이 때문에 중국사에 출현했던 전체 왕조에 대해서 중복되지 않도록 구분해 주어야 한다. 문제는 梁國(漢朝)과 같은 藩國은 중국사에서 그리 널리 알려져 있지 않기 때문에 확인하기 어렵다는 점이다. 이러한 왕조는 발견될 때마다 인스턴스를 추가하는 수밖에 없다.

211) 오대십국 중 하나인 朱全忠이 세운 後梁은 朱全忠의 성을 붙여 ‘朱梁’이라고도 부른다.

표 32. 중국에서 건국된 梁나라

번호	王朝	구분
1	梁國(春秋)	春秋시대 嬴姓의 諸侯國으로 지금의 陝西 韓城 일대에 존재했다.
2	魏國(戰國)	戰國시대의 諸侯國으로 大梁에 도움을 후에 梁國으로 칭했다.
3	梁國(漢朝)	漢왕조 시대의 藩國
4	梁國(晉朝)	晉왕조 시대의 藩國
5	南朝梁	南北朝 시대의 남조의 왕조로 梁武帝 蕭衍이 건국하였다.
6	西梁	梁武帝의 손자인 蕭詧이 세운 왕조
7	梁	隋末 唐初 蕭銑이 세운 왕조
8	梁	隋末 唐初 梁師都가 세운 왕조
9	梁	唐 초기에 沈法興이 세운 왕조
10	後梁	五대의 첫 번째 왕조로 朱溫(朱全忠)이 세운 왕조
11	五代十國時期	947年 契丹의 蕭翰이 後唐 明宗 李嗣源의 幼子 李從益을 옹립하여 세운 왕조

바. 사건(Event)

일반적으로 사건은 어떤 일이 일어나는 것을 지칭하나 확정적인 엄밀한 정의는 존재하지 않는다. CIDOC/CRM에서는 사건을 객체(thing)들이 시공간의 어떤 위치에서 만나는 것으로 정의하였으며, 개체와 개체 간의 작용과 반작용(Action Reaction)으로 정의하였다.²¹²⁾

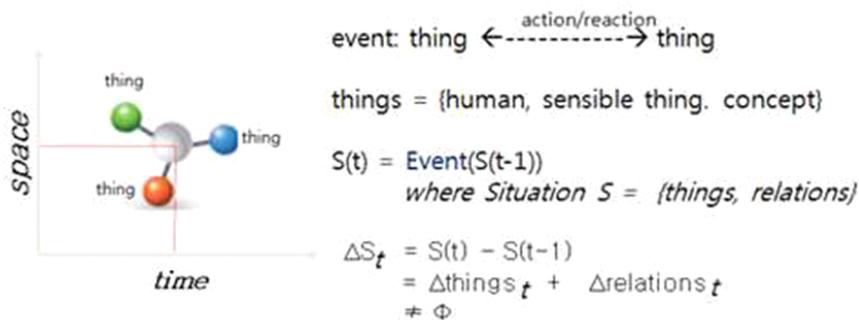


그림 36. 사건의 개념

212) 공간정보연구원, 대한지적공사, 「역사문화지도 구현을 위한 PROTOTYPE 개발 (최종보고서)」, 2014, 79쪽.

결국 사건은 기본적으로 무엇인가 일어나거나 발생하는 것을 지칭한다. 권창희는 역사문화적 상태는 이러한 개체들과 이들의 관계로 설명할 수 있으며, 사건은 상태를 변화시키고 상태의 변화는 개체 및 관계의 변화를 의미한다고 하였다.²¹³⁾ 그리고 사건 온톨로지를 EventID, 대표어(USE), 유의어(Synonym), 외국어, 주기(Scope Note), 시간위치(Temporal Location), 상위 사건(BT), 하위 사건(NT), 시간적 선후 관계, 공간위치(Spatial Location), 출전(Reference), 사건 분류 체계(Lineage), 관련 자료 등으로 구분하였다.²¹⁴⁾

조명대는 『조선왕조실록』을 대상으로 하여 사건을 ‘사건_실록명, 사건_실록관계기사, 사건_일으킨자, 사건연루_고위직, 사건_관계_인물, 사건_반대자, 사건_피해자, 사건_장소’ 등으로 구성하였다.²¹⁵⁾

이인근 등은 유럽에서 VICODI project²¹⁶⁾를 통해 유럽의 역사적 지식을 온톨로지로 표현하려는 연구가 수행되었으며,²¹⁷⁾ 타 분야의 지식과 달리, 역사적 사실의 전개는 주로 사건(event)에 기반하여 기술되고 있다고 하였다. 그리고 역사에서 하나의 사건은 주체, 시간, 대상, 장소, 행위 등과 같이 개념들이 복잡한 의미 관계를 형성하고 있으며, 사건이 일어나는 원인이나 특정 사건에 의해 발생하는 결과의 표현을 위해서는 역사적 지식이 사건 중심으로 표현되어야 한다고 하였다.²¹⁸⁾

본 논문에서도 사건에 대한 기본 개념은 객체(thing)들이 시공간의 어떤 위치에서 만나는 것이며, 개체와 개체 간의 작용과 반작용(Action Reaction)이라는 정의에 전적으로 동의한다. 다만 사건의 범위를 좀 더 확대하여 개체 간에 발생하는 행위나 상태까지 확대했을 뿐이다. 그렇기 때문에 본 연구에서의 사건은 계층을 이루고 있는 것으로 설계하였다.

이를 구체적으로 『通鑑節要』을 통해서 알아보자. ‘楚漢戰爭, 甘露의 變故, 黨錮의 禍, 玄武之變, 黃巢의 亂’ 등은 우리가 일반적으로 알고 있는 事件의 범주에 속한다. 이러한 事件은 다시 단기적인 것과 장기적인 것으로 구분할 수 있다. 皇帝의 卽位, 王朝의 건립, 馬嵬兵變²¹⁹⁾, 甘露의 變²²⁰⁾ 등의 사건은 지속된 구간이 짧다고 한다면, 楚漢戰爭²²¹⁾, 黃巢의 亂 등

213) 권창희, 「U-City에 있어서 U-Service의 콘텍스트 기반 역사문화 온톨로지 설계 관한 연구」,

『디지털정책연구』, 제9권, 제6호, 2011, 384쪽.

214) 권창희, 위의 논문, 제9권, 제6호, 2011, 384쪽.

215) 조명대, 「조선왕조실록 Linked Data 발행에 관한 연구」, 2012, 135쪽.

216) VICODI project, <http://www.vicodi.org>

217) 이인근 외, 위의 논문(2013), 156쪽.

218) 이인근 외, 위의 논문(2013), 156쪽.

219) 唐 玄宗의 아들 李亨이 陳玄禮 등을 포섭하여 馬嵬驛에서 군사들을 동원하여 楊國忠을 죽이고, 楊貴妃가 자살하도록 한 사건.

220) 唐나라 文宗 때 宰相 李訓·鄭注 등이 宦官 仇士良을 죽이려고 하늘에서 甘露가 내렸다고 속여 그들을 끌어내려 하다가, 목적을 달성하지 못하고 도리어 被殺당한 사건.

221) 기원전 206년부터 기원전 202년까지 4년 동안 楚나라와 漢나라가 천하를 놓고 벌인 전쟁으로

은 진행된 시간이 상당히 긴 사건이다. 이처럼 『通鑑節要』에 기록된 다양한 역사적 사건을 한 가지 레벨의 기술문으로 표현하기에는 어려움이 있다.

하나의 큰 사건은 그 사건을 구성하는 작은 사건이 있다. 사건을 중요성에 따라서 분류할 수도 있겠지만 그럴 경우 사건의 중요성을 객관적으로 판단하기란 쉽지 않다. 그렇기 때문에 본 논문에서는 사건의 시작 시점과 종료 시점 간의 기간의 길고 짧음으로 구분하였다. 이러한 지속 구간이 긴 사건에는 그 내부에 여러 개의 작은 사건이 존재하며 그 작은 사건에는 구체적인 행위가 존재한다. 이를 상위 사건과 하위 사건으로 구분할 수 있으며 행위는 인스턴스로 분류할 수 있다. 정리하면 상위 사건은 시간의 기간이 긴 사건을 말하며, 하위 사건은 지속된 시간이 짧은 사건을 말한다. 이들은 hasSubEvent와 hasSuperEvent 프로퍼티를 이용하여 상하 관계를 구분해 준다. 예를 들어 ‘楚漢戰爭’이라는 큰 사건은 ‘巨鹿大戰²²²⁾, 彭城戰鬪²²³⁾, 垓下之戰²²⁴⁾ 등의 작은 사건이 있으며, 이들 작은 사건 내에는 구체적인 행위가 있다. 구체적인 행위는 ‘공격하다, 죽이다, 점령하다, 대파하다’ 등과 같은 인간의 행위를 나타내는 동사가 기본 구성 요소가 된다.

사. 시간(Time)

시간 정보는 크게 시점과 기간(Interval)으로 구분할 수 있다. 기간은 다시 年, 月, 日처럼 시작 시점과 끝나는 시점이 일정하고 반복적인 기간과 그렇지 않은 기간으로 구분할 수 있다. CIDOC-CRM²²⁵⁾에 의하면 시간과 관련이 있는 엔티티는 E52 Time-Span(시기, 시작과 끝의 명시된 기간)와 E4 Period(시대)이다. ‘Time-Span’은 ‘1961, From 12-17-1993 to 12-8-1996, duration of the Ming Dynasty’처럼 일정한 시점과 종점이 있으며, ‘Period’는 ‘전사 시대 또는 청동기 시대, 오대 십국 시대, 고려 시대, 조선 시대’처럼 추상적인 시간 범위를 가진다. 본 논문에서도 年, 月, 日, 時처럼 시점과 종점이 명확한 경우에는

垓下 전투에서 項羽가 패해서 자살하게 됨으로써 한나라의 승리로 끝나게 된다.

222) 기원전 207년 楚나라의 項羽가 秦나라 章邯의 군대를 巨鹿(지금의 하북성 평향현 서남)에서 크게 물리친 전쟁을 말한다.

223) 彭城戰鬪는 기원전 203년 項羽가 楚나라 彭城을 점령한 劉邦의 56만 대군을 거의 전멸시키다시피한 전투를 말한다. 劉邦은 項羽가 齊 지방의 반란을 진압하기 위해 山東으로 가 있는 동안 56만의 대군을 동원하여 楚의 도성인 彭城을 점령한다. 이에 項羽는 3만의 군사를 이끌고 劉邦의 군대를 새벽에 급습하여 만나질도 안되는 시간에 56만의 군사를 궤멸시켰다.

224) 楚나라 項羽와 漢나라 劉邦이 垓下에서 벌인 마지막 戰鬪로 項羽가 韓信, 劉邦, 彭越, 黥布 등의 공격을 받아 패배하고 烏江가에서 자결하고 만다.

225) CIDOC CRM, ICOM/CIDOC, <http://www.cidoc-crm.org/Version/version-6.2>

‘Time-Span’이라고 하였으며, 불규칙적인 기간은 ‘Period’로 표현하였다. ‘Time-Span’은 다시 ‘PosthumousName, PosthumousNameYear, EraName, EraNameYear, TempleName, TempleNameYear’으로 하위 클래스를 분류하였다. 시호(PosthumousName)를 Time-Span 클래스의 하위 클래스로 설계한 것은 황제의 諡號는 사람을 의미하기도 하지만, 일정한 시간을 지칭하기 때문이기도 하다. 마찬가지로 ‘PosthumousNameYear’는 ‘太宗1년’처럼 일정한 기간을 표현한다. ‘Date’ 클래스는 하위 클래스로 ‘Day, Month, Year, SexagenaryCycle’을 두었으며, 인물의 행위가 발생한 시간과 종료된 시간, 사람의 태어나고 죽은 날, 사람이 관직에 나간 일이나 수상일 등을 연, 월, 일, 간지 등으로 표현할 수 있다.

시간과 관련된 쟁점은 크게 시간이라는 요소를 클래스로 분류해야 할지 아니면 리터럴로 처리해야 할지 결정해야 하는 문제와 일정한 기간을 가지고 있는 시간의 선후 관계를 어떻게 표현한 것인가이다. 먼저 시간 정보를 클래스로 처리할지, 리터럴로 처리할지를 결정해야 하는 이유는 시간을 클래스의 인스턴스로 분류할 경우에 Object Property로 관계를 맺어 줄 수 있기 때문에 시간을 통해서 자유로운 확장이 가능하다. 반면에 시간을 리터럴로 처리할 경우에는 자원 간의 관계를 Datatype Property로 맺어 주어야 하며, 시간을 통해서 새로운 노드의 전개가 불가능하기 때문이다. 시간의 선후 관계에 대해서는 뒤에서 프로퍼티에서 별도로 다룬다.

표 33. 시간 클래스 정의

클래스		내용	
Time	Day	일	
	Month	월	
	Year	년	
	SexagenaryCycle	간지	
	Period	시대(고려 시대, 조선 시대)	
	Time-Span	일정한 기간을 가지는 시간	
		PosthumousName	시호, 일정한 기간을 가지고 있음
		PosthumousNameYear	시호와 연도가 결합하여 기간을 표현
		eraName	年號가 가지는 기간을 표현
		eraNameYear	연호와 연도가 결합하여 기간을 표현
		TempleName	묘호, 일정한 기간을 가지고 있음
	TempleNameYear	묘호와 연도가 결합하여 기간을 표현	

아. 제도(System)

제도에는 여러 가지 클래스가 있을 수 있다. 여러 가지 제도 중에 과거 시험과 관련하여 'ExaminationPasser'와 'Examination'라는 클래스로 분류하였다. 클래스명을 'ExaminationPasser'를 둔 이유는 과거와 과거 시험에 합격자의 관계를 엄밀하게 따지면 과거 시험과 과거 시험에 합격자와의 관계는 정확한 계층 구조가 아니기 때문이다. 예를 들어 中宗甲午式年試²²⁶⁾에 합격한 사람에는 '金希聖²²⁷⁾, 丁應斗, 金三宅' 등이 있다. 이때 과거 시험과 과거 합격자 간의 관계를 표현하면 '中宗甲午式年試 hasInstance 金希聖'이라고 할 수 있다. 그렇다면 '金希聖'이 '中宗甲午式年試'을 구성하는 요소인지 살펴 보면, 관계는 있으나 구성요소로 보기는 어렵다. 왜냐하면 과거 시험을 구성하는 요소는 시험 합격자, 시험관, 시험 응시자, 試券 등이 올 수 있기 때문에 'hasInstance' 관계로 설명하기에는 명칭이 적절치 않다. 이 때문에 'ExaminationPasser'라는 클래스를 설정하고 하위 클래스로 각각의 방목을 두었다. 개별적인 과거 시험명은 'Examination'에 인스턴스로 등록한다. 한가지 아쉬운 점은 본 대상 자료가 『通鑑節要』인 점을 감안하면 중국의 과거제도에 맞는 클래스를 추가했어야 하는데 그러지 못하고 우리나라의 과거 제도를 중심으로 설명하였다. 이 부분은 좀 더 연구를 통해서 확장해 나갈 예정이다.

226) 中宗 29년(1534) 甲午 式年文科

227) 金希聖은 中宗 29년(1534) 甲午 式年試에 甲科 1등으로 합격하였다.

표 34. system

클래스		내용	
System		제도, 체제	
	Examination	과거 시험	
	ExaminationPasser	과거 합격자	
		ClassicsLicentiateExamination	생원진사시 합격자
		EruditeExamination	문과 합격자
		MilitaryExamination	무과 합격자
		MiscellaneousExamination	잡과 합격자

자. 어휘

한국, 중국, 일본, 대만, 베트남 등의 나라를 보통 한자 문화권이라고 한다. 이들 나라의 공통점은 현대 이전의 시대에 자국의 역사를 한문으로 기록했다는 것이다. 우리나라만 하더라도 『조선왕조실록』, 『승정원일기』 등이 모두 한문으로 씌여있다. 이러한 한문은 지식인의 필수 조건이었기 때문에 조선 시대 양반은 기본적으로 한문을 배워야 했다. 그러나 이제 시대가 변해 현대 사회에서는 소수의 전문가를 제외하고는 더 이상 한문을 학습하거나 배우기를 요구하지 않는다. 왜냐하면 한문은 이미 죽어 있는 언어이며 우리 시대가 더 이상 한문을 공부할 것을 요구하지 않기 때문이다. 이런 한문은 몇몇 호사가를 제외하고는 새롭게 만들어 지지 않고 있다. 그런데 만약 역사를 교육하거나 이를 이용하여 새로운 콘텐츠를 생산해야 할 경우에는 대부분 한문이 아닌 번역본을 이용한다. 이러한 이중적인 구조는 한문을 이해할 수 없기 때문에 당연한 것이다. 이런 현실적인 이유로 인해 한문 번역에 대한 수요는 계속 있어 왔으며 지금도 계속되고 있다. 『通鑑節要』를 예를들면 19세기 초에는 『詳密註釋 通鑑諺解』가 간행이 되었으며, 현대 들어 와서는 김충렬의 『資治通鑑』天·地·人²²⁸⁾, 성백효의 『역주 통감절요』1~9, 김정화의 『통감절요』1~4²²⁹⁾, 이충구의 『(국역) 통감절요증손교주』1~3²³⁰⁾ 등이 계속해서 간행되었다. 그런데 언어라는 것은 시대의 흐름에 따라 변하기 마련이다. 19세기 초에는 언해본을 통해서 원문에 대한 이해를 하였다면, 근래에는 요즘 언어에 맞는 새로운 번역본을 찾아서 활용하고 있다. 그리고 이런 현상은 앞으로도 계속될 것

228) 江贊(지)/김충렬(역), 『資治通鑑』天·地·人, 三省出版社, 1987.

229) 江贊(지)/김정화(역), 『통감절요』1~4, 충북대학교 출판부, 2015.

230) 江贊(지)/이충구·박동현·한윤숙·황봉덕·선미현·김규선·윤세형·백호현·서은선·이승용·

김현재·김보성·강민정·권치은·시습학사, 『(국역) 통감절요증손교주』, 다운샘, 2016.

이다.

한편 2017년 한국고전번역원에서는 승정원일기 번역에 소요되는 시간을 단축하기 위해 인공지능 기반 고전문헌 자동번역시스템을 개발하고 있다.²³¹⁾ 고전번역원은 병렬 말뭉치를 구축하여 신경망 번역 방법을 이용하여 자동 번역을 시도하고 있다. 2011년에 왓슨은 미국의 퀴즈 프로그램 ‘쥬파디!(Jeopardy!)에 출현하여 역대 챔피언과 대결해 승리하면서 상금 100만달러(약 10억원) 획득하면서 일약 각광을 받았다. 왓슨은 위키피디아의 기술을 바탕으로 라이트웨이트 온톨로지를 생성하고, 그것을 정답에 사용하였다.²³²⁾ 라이트웨이트 온톨로지는 컴퓨터에 데이터를 읽게 해서 자동으로 개념 간의 관계성을 생성하는 것을 말한다.

이런 시대적인 요구에 맞춰 역사적 지식도 한문이라는 장벽을 넘어야 한다. 아직은 요원하지만 한문에 사용된 동사나 형용사를 중심으로 의미역을 자동으로 파악하고, 각 개체 사이를 자동으로 연결해 줄 수 있다면, 한문을 전체를 읽지 않아도 그 내용을 자동으로 이해할 수 있을 것이다. 본 연구의 목표인 역사 자료의 연계 및 통합과 지식 관계망에는 이러한 언어 간의 연계, 자료 간의 연계를 포함한 개념이다. 지식 관계망에는 어휘 간의 상하 관계뿐만 아니라 여기서 설명하는 대역어도 중요한 역할을 한다.

과거에 자동 번역은 어휘 관계망을 통해서 추진되어 왔다. 울산대학교나 부산대학교 등에서 구축한 어휘망을 그러한 용도로 사용할 수 있을 것이다. 해외에서는 이러한 노력이 훨씬 일찍부터 시작되었다. 유로 워드넷은 기계 번역, 정보검색, 자동 요약 등을 위해 광범위한 어휘 의미에 대한 분석을 시도하고 있다. 그런데 자동 번역을 위해서는 병렬 말뭉치를 기반으로 하던 어휘망을 기반으로 하던 간에 한문의 번역문제는 쉽게 해결되지 않을 것이다. 일단 대상이 되는 한문으로 된 말뭉치가 절대적으로 부족하고, 어휘망은 대부분 현대 한국어를 대상으로 하고 있기 때문에 고전을 이해하는데로 큰 도움이 되지 않기 때문이다. 더욱이 이러한 노력은 대부분 정제화된 어휘에 대해서 진행하고 있으며 번역에 사용되는 구체적인 사례를 개념화하지는 않고 있다.

이때문에 한문 원전으로 쓰여 있는 역사 사료에 사용된 어휘가 현대어로 번역되었을 때 어떻게 번역되는지 이들 어휘 간의 상관 관계에 대해서도 지속적으로 탐구할 필요가 있다. 그 이유는 번역되는 어휘가 비록 시대에 따라 사용되는 표현은 달라지지만 그 본질적인 의미를 달라지지 않기 때문이다. 예를들어 아래 문장에서 황제의 죽음을 뜻하는 崩을 ‘崩하고’로 번역하기도 하였고, ‘승하하다’라는 번역한 경우도 있었다.

231) 2017년 5월부터 정부의 지원으로 ‘인공지능 기반 고전문헌 자동번역시스템 구축(조달청

입찰공고번호 : 20170417811-00)’ 사업을 진행하고 있다. 이 사업의 제안요청서를 보면 개발된

번역 시스템이 자동 번역한 결과물을 1차 초벌 번역에 활용하는 것을 목표로 삼고 있다.

232) 마쓰오 유타카(저)/박기원(역), 『인공지능과 딥러닝』, 동아엠앤비, 2015, 102쪽.

■ 三月에 帝崩_고 宣宗이 卽位_{하다}.

☞ 三月에 帝 | 崩_고 宣宗이 位에 卽_{하다}(『詳密註釋 通鑑諺解』)

☞ 3월에 황제가 승하_고 宣宗이 즉위하였다.(『역주 통감절요』)

또한 어휘에 대한 정리는 정제된 어휘만을 계층적으로 정리하는 것으로는 다 해결할 수 없을 것이다. 한 언어를 다른 언어로 번역할 때 원문의 의도와 느낌을 정확하게 전달할 수 있는 적절한 어휘를 선택할 때 도움을 줄 수 있는 것은 기존에 번역한 다양한 사례이다. 이런 번역 사례는 기존에 번역된 성과를 가장 효율적으로 흡수할 수 있는 현실적인 방안일 것이다. 이점을 염두에 두고 본 연구에서는 원문과 번역문에 사용된 어휘를 다음의 표와 같이 모두 1:1로 연결하여 활용할 수 있도록 하였다.

표 35. 원문 번역문 대역어 예시

구분	원문	번역문	원문	번역문	원문	번역문	원문	원문	번역문
ID	event_1	event_1	event_2	event_2	event_3	event_3	event_4	event_5	event_5
AG	李德裕	李德裕	唐宣宗李忱	唐宣宗李忱	唐武宗李炎	唐武宗李炎	李德裕	白敏中	白敏中
AC	引	추천하다 02	爲	임명하다 01	崩	승하하다	失	乘	틈타다
AL									
CA									
CO									
CT									
DE									
DR									
DS									
EF									
FR									
GE									
GO			翰林學士	翰林學士					
IN									
LO									
PA	白敏中	白敏中	白敏中	白敏中					
QU									
RO									
RT									
SR									
TE							勢力	上下之怒	上 下 의 노여움
TM	初								
DA					846.4.22	846.4.22	847	847	847
year					846	846	847	847	847

이들 어휘는 모두 Object Property인 ‘translationWord’를 이용하여 원문과 번역문이 상호 연결을 한다. 대역어라는 것이 특정 어휘만을 의미하는 것이 아니라 본문에 나와 있는 전체 어휘를 의미하기 때문에 정의역과 치역이 모두 Thing이 된다. 그런데 이들 어휘는 항상 정해진 대로 번역되는 것이 아니라 그때그때 상황과 문맥에 따라서 다르게 번역되기 때문에 번역된 구체적인 사례를 수집하는 것이 무엇보다도 중요하다. 그리고 하나의 어휘는 여러 개의 대역어와 연결이 되며, 반대의 경우도 마찬가지이다. 다음의 그림에서 보듯이 誅라는 동사가 ‘주벌하다, 죽이다_01, 죽다_01’ 등 8개로 세분화되었으며, 역으로 ‘죽다_01’의 의미는 한자어가 ‘崩, 死, 殂’ 등 6개의 어휘가 연결이 되었다. 이런 어휘의 구체적인 사례가 쌓임으로서 자동 번역을 할 수 있는 기초가 갖추어 지는 것이다.

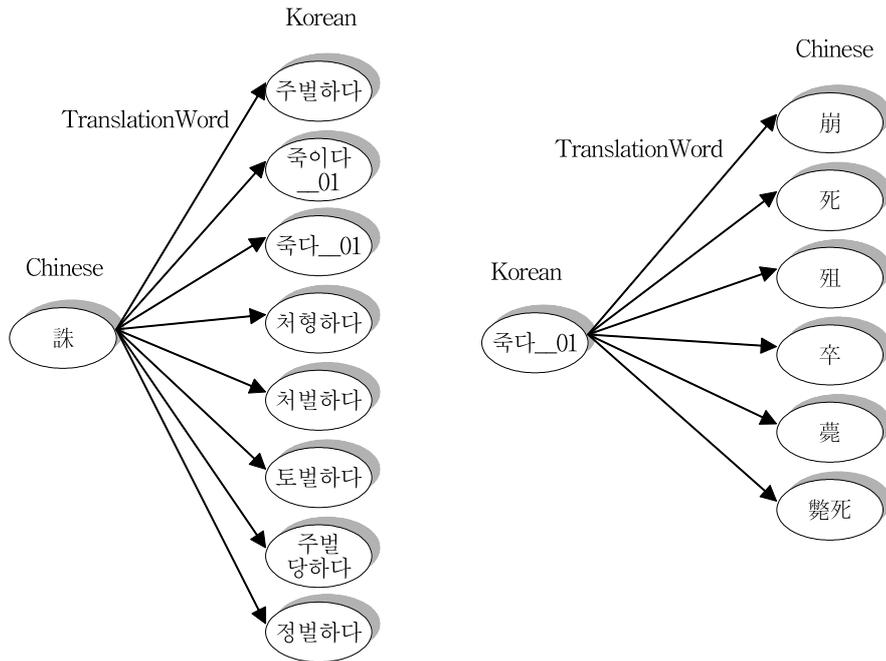


표 36. 대역어의 RDF 그래프 표현

어휘 간 연계를 위해서는 event를 비롯한 모든 어휘가 식별되어야 한다. 이때 어휘를 식별하는 가장 좋은 방법은 한 글자로 된 漢字는 현재까지 가장 상세한 뜻 풀이가 되어 있는 『漢韓大辭典』의 식별 번호를 따르는 것이고, 두 개 이상의 漢字로 구성된 한자 어휘는 전통문화연구회의 동양고전번역지원 시스템에서 제공하는 ID를 따르는 것이다.²³³⁾ 이 시스템에는 『漢韓大辭典』을²³⁴⁾ 비롯하여 『표준국어대사전』, 『漢語大詞典』, 『大漢和辭典』 등에 수록된 표제어를 수집하여 출처 정보와 함께 약 150만 개의 어휘를 제공하고 있다. 아직은 전체 어휘에 대한 통합 작업을 이루어지지 않았으나 『漢韓大辭典』의 식별자는 확인할 수 있으므로 이 식별자를 따른다.

233) 동양고전번역지원, 전통문화연구회, <http://lu.juntong.or.kr/juntong>

234) 단국대학교 동양학연구소에서 30년 간 310여 억원을 들여 완성한 세계 최대 규모의 한한사전으로 모두 16권으로 구성되어 있으며, 약 6만여 자의 한자와 45만 여 개의 어휘를 수록하고 있다.

표 37. 대역어 연계를 위한 클래스

클래스		내용
thing		어휘
	Korean	한국어 어휘
	Chinese	한문 어휘

차. 기타

1) 인공물(Artifacts)

인공물은 사람의 창작물을 포함한다. 하위 클래스로 ‘Book, Document, Works’로 구분하였다. ‘Document’는 고문서 등의 장정이 되지 않는 문서 형태를 의미하며, 작품은 문학 작품, 예술 작품 등을 포괄하는 의미로 사용하였다.

표 38. Artifacts

클래스		내용
Artifacts		인공 구조물, 건물, 시설
	Book	서명
	Document	문서
	Works	작품

다음은 고사성어 클래스이다. 『通鑑節要』에는 많은 고사성어가 등장한다. 우리 선조들은 어려서부터 『通鑑節要』를 배우다 보니 자연스럽게 문집 등에서도 『通鑑節要』의 고사를 인용하는 경우가 많았다. 고사성어는 Idom으로 표현하고 풀이말과 description 프로퍼티를 사용하여 문자열과 연결한다. 나머지 동물(animal), 식물, 천문현상, 개념용어, 정의, 왕조, 신분, 사물, 관직, 관서, 상훈 등은 별도로 설명하지 않는다. 지금까지 분석한 클래스를 정리하면 다음과 같다.

표 39. 클래스

번호	클래스	내용	
1	Action	행위, 구체적인 행위	
2	Agent	Faction	당파, 파벌
		Family	가문
		Group	집단, 조직, 당파, 학파
		Organization	기관, 단체
		Person	인물
		Tribe	종족, 씨족, 부족, 종족, 민족
3	Animal	동물	
4	Artifacts	Book	서명
		Document	문서
		Works	작품
5	AstronomicalPhenomenon	천문 현상	
6	Concept	개념	
7	Description	어휘에 대한 간단한 설명	
8	Dynasty	왕조명	
9	Event	사건	
10	Idiom	고사성어	
11	Object	일반적인 의미의 사물	
12	Office	관서	
13	Place	지명, 행정 지명, 자연 지명, 인문 지명	
14	Plant	식물	
15	Positon	관직, 전통 시대의 관서, 조직	
16	Prize	포상, 넓은 범위에서 포상	
17	Religion	종교	
18	Source	출처 정보	
19	Status	신분, 법적인 신분	
20	System	법, 제도	
		Examination	과거
		ExaminationPasser	과거합격자
21	Term	용어	
22	Time	Date	날짜
		Interval	기간
		Period	시대
		Timespan	기간

4. 프로퍼티의 설계

OWL에서 개체 간의 관계 혹은 개체와 리터럴 간의 관계는 프로퍼티(Property)라는 속성

으로 표현한다. 이 속성은 다시 오브젝트 프로퍼티(ObjectProperty)와 데이터 타입 프로퍼티(DatatypeProperty)로 구분한다. 오브젝트 프로퍼티(ObjectProperty)는 클래스 내의 인스턴스²³⁵⁾ 간의 관계를 표현하며, 데이터 타입 프로퍼티(DatatypeProperty)는 인스턴스가 취해야 하는 데이터의 형식과 값을 표현한다. 프로퍼티에는 주어로 올 수 있는 것과 목적어로 올 수 있는 것, 제약 조건 등을 명시해야 한다. 클래스가 의미 간의 계층화가 중요하다면, 프로퍼티는 클래스에 비해 속성을 잘 표현할 수 있는 어휘의 선택이 특히 중요하다.²³⁶⁾

1. 通鑑節要 creator 江贄 ⇨ 通鑑節要의 저자는 江贄이다
2. 江贄 create 通鑑節要 ⇨ 江贄가 通鑑節要를 짓다

1과 2는 『通鑑節要』를 江贄라는 사람이 지었다'라는 사실을 전달한다는 측면에서 형식은 약간 다르지만 모두 거의 동일한 의미를 전달한다고 할 수 있다. 1은 관계를 creator라는 명사로 표현하였으며 2는 create라는 동사로 표현하였다.²³⁷⁾ RDF 트리플에서 서술어를 명사로 표현할 경우에 의미 전달이 혼란스러울 수 있으므로 가급적이면 서술어 형태로 명명하는 것이 좋다. 아래는 SKOS(Simple Knowledge Organization System)에서 사용하는 온톨로지 어휘 중 하나인 'skos:broader'의 예이다.

cheese skos:broader dairy

위의 문장은 'cheese'는 'dairy'라는 광의어를 가지고 있다 혹은 'cheese'의 광의어는 'dairy'이다로 읽어야 하나 'cheese'는 'dairy'보다 광의어이다로 잘못 읽을 수 있다. 이런 경우에 skos:broader²³⁸⁾에 대해 'rdfs:label'을 적절히 사용하여 'skos:broader rdfs:label "has broader term"'으로 명시하여 의미를 분명하게 전달함으로써 어휘를 오해하는 혼란을 완화시킬 수 있다.²³⁹⁾

가. 개체 간 관계의 표현

프로퍼티는 개체와 개체 간의 관계를 표현하는 것이다. 역사서에 나타나는 프로퍼티를 파악하기 위해서 가장 좋은 방법은 사전에 정리된 프로퍼티를 이용하여 직접 적용해 보는 것

235) 인스턴스(Instance)는 Individual이라고도 한다. Individual은 'Instances of Classes(클래스의 한 개체)'로 말하기도 한다.

236) 물론 클래스 또한 개체의 집합을 가장 잘 표현할 수 있는 이름을 짓는 것이 중요하지 않다는 의미는 아니다. 프로퍼티에서 특히 이름을 짓는 것이 클래스에 비해 더 중요하다는 의미이다.

237) 더블린코어 메타데이터를 온톨로지로 표준화하면서 이전에 사용하던 이름을 그대로 사용하였다.

238) W3C, SKOS Simple Knowledge Organization System Primer, 2009,

<https://www.w3.org/TR/2009/NOTE-skos-primer-20090818>

239) Dean Allemang·James A. Hendler(저)/김성혁 외(역), 앞의 책(2008), 297쪽.

이다. 실제로 적용하다보면 생각지 못했던 여러 가지 관계와 문제점이 드러나기 마련이다. 역사서에 출현하는 인물, 지명, 사건 등의 개체 중에 가장 중요한 것은 역사의 주체가 되는 인물이다. 이 사실은 『通鑑節要』 본문 중에 출현하는 개체명을 분석하였을 때 인명이 압도적으로 많은 것에서 확인할 수 있었다. 다음으로 사람과 사람의 관계를 살펴보면 혈연 관계, 혼인 관계, 사회적 관계 등이 나타날 수 있다. 이러한 관계 유형을 정리하기 위해서 코어넷(CoreNet)과 CBDB(China Biographical Database), CIDOC-CRM(CIDOC Conceptual Reference Model)을 분석해 본다.

1) 코어넷(CoreNet)

한국과학기술원(KAIST) 전문용어언어공학연구센터(KORTERM)에서는 컴퓨터 언어 처리를 위하여 약 10년 전부터 구축해 온 다양한 코퍼스와 전자사전을 기반으로, 자연언어처리 및 지식 처리 과정에서 발생하는 의미 모호성을 해소하기 위해 다국어 어휘 의미망(CoreNet)을 구축하였다.²⁴⁰⁾ 코어넷에서는 사람과 사람의 관계를 다음과 같이 친족관계, 대인관계, 교제관계, 동료관계, 동지관계 등으로 구분하였다.

240) 코어넷, KAIST, http://semanticweb.kaist.ac.kr/org/bora/CoreNet_Project/index.html

표 40. 코어넷의 인간 관계

개념번호	개념범주	단어
#1111	사람	-
#11111	인간	-
#111111	인칭	-
#111112	인간(생물학적 특징)	-
#111113	인간(친족 관계)	-
#111114	인간(대인 관계)	-
#1111141	인간(교제 관계)	-
#11111411	동료/일원/방외인	-
#111114111	동료/일원	-
#1111141111	동지	관료, 남지, 단짝, 동급생, 동기, 동기생, 동료, 동무, 동문, 동인, 동지, 동창생, 동학, 방배, 식구, 아군, 일단, 자파, 한패, 한편
#1111141112	신도	교도, 교인, 권사, 그리스도인, 당도, 당기, 대사, 도가, 도반, 도사, 도인, 독신자, 소금, 순교자, 신자, 양 여관, 이슬람 자매, 지지자, 명사, 크리스천, 평신도, 형제, 회교도
#1111141113	동반자	관행, 동반자, 동행, 반려, 일행

2) CBDB(China Biographical Database)

CBDB(中國歷代人物傳記資料庫, CBDB)는²⁴¹⁾ 唐나라부터 淸나라까지 중국의 주요 인물의 개인 전기 자료를 모아서 정리한 데이터베이스이다. CBDB는 미국의 宋代史 연구자 Robert Hartwell이 1980년대에 일일이 손으로 정리한 20,000여 명의 宋代 인물 전기 자료를 Michael Fuller 교수가 MS-엑세스로 변환한 것이다. 그 이후 미국의 하버드대학, 中國 北京大學古代史研究中心, 타이완 中央研究院歷史語言研究所 간의 상호 협력을 통해 발전, 확대시키고 있다. CBDB는 2017년 4월 기준으로 약 360,000명의 인물에 대한 자료를 정리하였으며, 누구나 제약 없이 사용할 수 있도록 공개되어 있다.²⁴²⁾

CBDB는 각 인물의 정보를 관계형 데이터베이스의 데이터 설계 규칙에 의거하여 인물의 기본 정보(출생 연도, 사망 연도, 출생지 등), 인맥 관계(혈연 관계와 사회 관계), 관직 정보 등으로 분류·정리하였다. CBDB에서 제공하는 인물 관련 정보는 본 논문의 지식 관계망 구축에도 많은 부분 포함될 수 있으므로 관계에 대해서 좀 더 깊이 있게 알아보기로 한다.²⁴³⁾

241) 中國歷代人物傳記資料庫, 하버드대학교, <http://projects.iq.harvard.edu/chinesecbdb/home>

242) 조원희, 「디지털 인문학과 元代史(원대사) 研究(연구) -中國傳記資料庫(CBDB)의 소개와 전망을 중심으로」, 『중앙아시아연구』, 제21권, 제1호, 2016, 119쪽.

243) CBDB를 활용하기 위해서는 다음의 두 가지를 고려해야 한다. 첫째, CBDB는 唐부터 淸까지의 인물 정보를 제공하고 있으나 왕조에 따라서 포함된 데이터의 품질과 양의 차이가 상당하다.

CBDB에서 사람과 사람의 관계를 살펴보면 관계는 ‘Friend of, Menren of’ 등과 같은 변화가 거의 없는 정적인 관계도 있지만, ‘彈劾, 反對/攻訐’ 등과 같이 동사로 표현되는 여러 가지 행위에 대해서 모두 관계로 표현한 것을 확인할 수 있다.

표 41. CBDB 인물 관계 예시

관계 영문	관계 중문
Patron of (= Client was)	是Y的恩主
Patron was (= Client of)	恩主是Y
Coalition member of	黨羽爲Y
Coalition leader of	黨魁爲Y
Friend of	友
Friend in the same graduating class	同年友
Impeached	彈劾
Impeached by	被Y彈劾
Recommended	推薦
Recommended by	被Y推薦
Opposed or attacked	反對/攻訐
Opposed by or attacked by	遭到Y的反對/攻訐
Praised or admired by	被Y欣賞/器重
Praised or admired	欣賞/器重
Menren of	爲Y之門人
Menren was	門人爲Y
Coalition associate of	與Y結黨
Student of	爲Y之學生
Student was	學生爲Y
Directly recruited (bi, zoubi, etc)	辟
Directly recruited (bi, zoubi) by	爲Y所辟
Supported by	得到Y的支持

CBDB에서 사람과 사람의 관계로 표현한 것을 살펴보면, 우선 혈연 관계와 혼인 관계 등과 같은 거의 변하지 않는 정적인 관계와 ‘감상하다, 탄핵하다’ 등과 같은 동적인 관계가 있다. 여기서 동적인 관계는 대부분 동사 혹은 형용사로 표현할 수 있는 관계이다. 이러한 예로는 ‘請罪, 反對, 棄市, 遠竄, 拿推, 斬, 殺害, 支持, 誣告’ 등이 있으며 대부분 한자어로 된 행위 명사라는 공통점이 있다. 행위 명사는 주로 ‘斬하다, 棄市하다, 支持하다’처럼 ‘하다’와 결합하여 행위 동사로 활용된다. ‘탄핵되다’처럼 ‘되다’와 결합하여 피동의 의미를 지니기도 한다.

둘째, CBDB 데이터베이스 내에 있는 정보 중에 적지 않은 오류가 있을 수 있다. 그 이유는 배포 원칙을 완벽하게 정제된 데이터를 배포하는 것이 아닌 텍스트 데이터 내에서 기계적 추출을 통해서 먼저 정리하고 공개한 후에 이후 점차 보완해 나가는 것을 목표로 하였기 때문이다.

3) CIDOC-CRM

CIDOC-CRM(CIDOC Conceptual Reference Model)은 국제박물관협회(International Council of Museums)의 문서화를 위한 국제 위원회(International Committee for Documentation)에서 개발한 문화유산 정보를 위한 개념 참조 모델이다.²⁴⁴⁾ 2017년 현재까지 Version 6.2.2까지 발행되었으며, 웹사이트(new.cidoccrm.org)²⁴⁵⁾를 통해 관련 기술 문서를 확인할 수 있다. CIDOC CRM은 문화유산과 박물관 유물에 대한 정보의 기록을 위해 개발된 객체 지향 온톨로지(object-oriented ontology)로서 개체명을 중심으로 도메인(domain) → 속성 관계(property) → 범위(range)로 구조화되어 있다.²⁴⁶⁾ 개체명은 모두 137개의 엔티티로 구성되어 있으며, 프로퍼티는 151개로 구성되어 있다.²⁴⁷⁾ 시간 엔티티(Temporal Entity), 지속성 항목(Persistent Item), 시간범위(Time-Span), 장소(Place), 치수(Dimension), 시공간 용량(Spacetime Volume) 등 6개 엔티티에 각기 하위 엔티티를 가지고 있다.

표 42. CIDOC CRM Class Hierarchy

E1	CRM Entity
E2	- Temporal Entity
E4	- - Period
E5	- - - Event
E7	- - - - Activity
E11	- - - - - Modification
E12	- - - - - - Production
E13	- - - - - - Attribute Assignment
E65	- - - - - - Creation
E63	- - - - - - Beginning of Existence
E12	- - - - - - Production
E65	- - - - - - Creation

244) CIDOC CRM, International Council of Museums, <http://www.cidoc-crm.org>

245) International Committee for Documentation (CIDOC). CIDOC CRM Special Interest Group, <http://www.cidoc-crm.org>

246) ICOM/CIDOC Documentation Standards Group, 2001

247) ICOM.documentation of collections> International Committee for Documentation (CIDOC). CIDOC CRM Special Interest Group., 「Definition of the CIDOC Conceptual Reference Model」, 2016.

E64 -	-	-	-	End of Existence
E77 -	Persistent Item			
E70 -	-	Thing		
E72 -	-	-	Legal Object	
E18 -	-	-	-	Physical Thing
E24 -	-	-	-	Physical Man-Made Thing
E90	-	-	-	Symbolic Object
E71 -	-	-	Man-Made Thing	
E24 -	-	-	-	Physical Man-Made Thing
E28 -	-	-	-	Conceptual Object
E89 -	-	-	-	Propositional Object
E30 -	-	-	-	Right
E73 -	-	-	-	Information Object
E90 -	-	-	-	Symbolic Object
E41 -	-	-	-	Appellation
E73 -	-	-	-	Information Object
E55 -	-	-	-	Type
E39 -	-	Actor		
E74 -	-	-	Group	
E52 -	Time-Span			
E53 -	Place			
E54 -	Dimension			
E59	Primitive Value			
E61 -	Time Primitive			
E62 -	String			

4) RELATIONSHIP 어휘스키마

RELATIONSHIP 어휘스키마는 Davis(2010)와 Vitiello(2010)가 2004년에 처음으로 인물과 인물 간 상세한 인간 관계를 기술하기 위해 작성한 어휘스키마이다. 2009년 5월 현재 34종의 어휘가 유지·관리되고 있다.²⁴⁸⁾ RELATIONSHIP 어휘 스키마는 앞 절에서 FOAF 어

248) RELATIONSHIP, <http://purl.org/vocab/relationship>

휘스키마의 객체속성 foaf:knows의 하위 객체 속성으로서 인물과 인물 사이에서 발생하는 다양한 인간 관계를 34종의 관계어휘를 사용하여 상세하게 표현할 목적으로 정의된 것이다.²⁴⁹⁾

표 43. RELATIONSHIP 온톨로지 어휘

Term	Description
Acquaintance Of	A person having more than slight or superficial knowledge of this person but short of friendship
Ambivalent Of	A person towards whom this person has mixed feelings or emotions
Ancestor Of	A person who is a descendant of this person
Antagonist Of	A person who opposes and contends against this person
Apprentice To	A person to whom this person serves as a trusted counselor or teacher
Child Of	A person who was given birth to or nurtured and raised by this person
Close Friend Of	A person who shares a close mutual friendship with this person
Collaborates With	A person who works towards a common goal with this person
Colleague Of	A person who is a member of the same profession as this person
Descendant Of	A person from whom this person is descended
Employed By	A person for whom this person's services have been engaged
Employer Of	A person who engages the services of this person
Enemy Of	A person towards whom this person feels hatred, intends injury to, or opposes the interests of
Engaged To	A person to whom this person is betrothed
Friend Of	A person who shares mutual friendship with this person
Grandchild Of	A person who is a child of any of this person's children
Grandparent Of	A person who is the parent of any of this person's parents
Has Met	A person who has met this person whether in passing or longer
Influenced By	a person who has influenced this person
Knows By Reputation	A person known by this person primarily for a particular action, position or field of endeavour
Knows In Passing	A person whom this person has slight or superficial knowledge of
Knows Of	A person who has come to be known to this person through their actions or position
Life Partner of	A person who has made a long-term commitment to this person's
Lives With	A person who shares a residence with this person
Lives With	A person who shares a residence with this person
Lost Contact With	A person who was once known by this person but has subsequently become uncontactable
Mentor Of	A person who serves as a trusted counselor or teacher to this person
Neighbor Of	A person who lives in the same locality as this person
Parent Of	A person who has given birth to or nurtured and raised this person
Participant	-
Participant In	-

249) 강현민, 「RDF/OWL의 객체속성을 이용한 관계온톨로지 시스템 구축과 활용에 관한 연구」, 연세대학교 대학원, 2010, 105쪽.

Relationship	A particular type of connection existing between people related to or having dealings with each other
Sibling Of	A person having one or both parents in common with this person
Spouse Of	A person who is married to this person
Works With	A person who works for the same employer as this person
Would Like To Know	A person whom this person would desire to know more closely

나. 오브젝트 프로퍼티(ObjectProperty)

1) 인물 관계(hasRelatedPerson)

가) 가족 관계(hasRelatedPerson)

개체와 개체 간에 형성되는 관계는 많지만 크게 의미적인 관계와 내용적인 관계로 구분할 수 있다. 의미적인 관계는 개체의 의미에서 발생하는 관계로 領議政-領相처럼 同義語, 類義語, 反意語와 같은 관계이다. 내용에서 발생하는 관계는 項羽-項籍처럼 어휘 자체의 의미가 아니라 『史記』라는 책에서 출현한 역사적 사실에 의해 발생하는 관계이자 개별적이고 구체적인 사례의 관계이다. 내용적인 관계는 변하지 않는 정적인 관계와 자주 바뀌는 동적인 관계로 구분할 수 있다. 앞서 언급한 項羽와 項籍의 관계는 시간의 흐름에 따라서 그 관계가 변하지 않는다. 그러나 한 사람이 역임한 ‘관직’이라면 관직은 시간의 흐름에 따라 자연스럽게 변할 수밖에 없다. 내용에 의해서 발생하는 관계의 대표적인 것은 인간과 인간의 관계이다. 인간 간 관계는 크게 혈연 관계, 혼인 관계, 사회적 관계로 구분할 수 있다. 이 중에서 혈연 관계나 혼인 관계에 대해서는 큰 이견이 없다. 그러나 사회적 관계는 아직까지 뚜렷하게 정리된 기준이 없다. 이러한 인물 간 관계는 지식 관계망을 구성하는 중요한 요소가 된다.

사람 간의 관계는 ‘hasRelatedPerson’으로 표현한다. 혈연 관계는 ‘Person’ 클래스와 ‘Person’ 클래스의 관계이기 때문이다. 하위 오브젝트 프로퍼티로 혈연관계와 사회적 관계를 ‘hasRelatedFamily, hasRelatedSocial’로 표현하였다. 조상과 부모의 계층 관계를 설정할 때 이행 관계의 적용 여부를 검토해야 한다. 조상을 표현하는 ‘hasAncestor’는 이행적이나 부모를 표현하는 ‘hasFather’나 ‘hasMother’는 이행적이지 않다. 즉 ‘나의 조상의 조상은 나의 조상’이라는 관계는 참이 되지만 ‘나의 부모의 부모는 나의 부모’라는 관계는 참이 되지 않는다. 이러한 관계를 고려하여 ‘hasAncestor’는 생부와 양부를 ‘hasFosterFather, hasBirthFather’로 구분하였으며 일반적인 아버지는 ‘hasFather’로 표현하였다.

■ hasAncestor

- ☞ A hasFather B : A는 B를 아버지로 가지다
- ☞ A hasBirthFather B : A는 B를 생부로 가지다
- ☞ A hasFosterFather B : A는 B를 양부로 가지다
- ☞ A hasMother B : A는 B를 어머니로 가지다
- ☞ A hasBirthMother B : A는 B를 생모로 가지다
- ☞ A hasStepMother B : A는 B를 계모로 가지다
- ☞ A hasFatherInLaw B : A는 B를 장인으로 가지다
- ☞ A hasMotherInLaw B : A는 B를 장모로 가지다

자손의 관계 표현은 적서의 구분과 친자와 양자, 친녀와 양녀를 구분해 주었다.

■ hasDescendant

- ☞ A hasSon B : A는 B를 아들로 두다
- ☞ A hasLegitimateSon B : A는 B를 적자로 두다
- ☞ A hasBastardSon B : A는 B를 서자로 두다
- ☞ A hasFosterSon B : A는 B를 양자로 두다
- ☞ A hasDaughter B : A는 B를 딸로 두다
- ☞ A hasLegitimateDaughter B : A는 B를 적녀로 두다
- ☞ A hasBastardDaughter B : A는 B를 서녀로 두다
- ☞ A hasFosterDaughter B : A는 B를 양녀로 두다

인척 관계는 처와 첩을 구분하였으며, 사위와 며느리가 인척 관계에 포함된다.

■ hasRelatedAffinity

- ☞ A hasWife B : A는 B를 부인으로 가지다
- ☞ A hasConcubine B : A는 B를 첩으로 두다
- ☞ A hasHusband B : A는 B를 남편으로 가지다
- ☞ A hasSonInLaw B : A는 B를 사위로 두다
- ☞ A hasDaughterInLaw B : A는 B를 며느리로 두다

형제 관계에는 형제와 자매의 관계를 표현한다.

■ hasRelatedSibling

☞ A hasBrothers B : A는 B를 형제 관계를 가지다

☞ A hasSisters B : A는 B를 자매 관계를 가지다

이들 관계를 전체적으로 정리하면 다음과 같다.

표 44. 인물 관계 프로퍼티

속성			내용
hasRelatedPerson			관계를 가지다
	hasRelatedFamily		가족 관계를 가지다
		hasAncestor	조상을 가지다
		hasBirthFather	生父
		hasFather	父
		hasFosterFather	養父
		hasBirthMother	生母
		hasMother	母
		hasStepMother	養母
		hasMotherInLaw	丈母
		hasFatherInLaw	丈人
		hasDescendant	자손을 가지다
		hasBastardDaughter	庶女
		hasBastardSon	庶子
		hasDaughter	女
		hasFosterDaughter	養女
		hasFosterSon	養子
		hasLegitimateDaughter	嫡女
		hasLegitimateSon	嫡子
		hasSon	子
		hasRelatedAffinity	婚姻關係
		hasConcubine	妾
		hasDaughterInLaw	며느리
		hasHusband	男便
		hasWife	婦人, 娶, 配
		hasSonInLaw	壻
		hasRelatedSibling	兄弟
		hasBrothers	男子 兄弟
		hasSisters	女子 姉妹

나) 사회적 관계(hasRelatedSocial)

가족 관계나 혈연 관계는 그 수가 제한적이라 프로퍼티로 표현하는 데 큰 문제가 없는데 비해 사회적 관계는 그 범위가 모호하고 나올 수 있는 경우가 매우 다양하기 때문에 직접적인 방식으로 표현하기 어렵다. 이혜원·윤소영은 기존의 한국역사용어시소러스의 개념 표현 기준과 전문가들이 작성한 마인드맵을 분석하여, 역사용어 온톨로지 모형을 설계하면서 인물들 간의 관계를 동지·우호 관계, 적대 관계로 분석하였다. 사회적 관계란 혈연 관계나 혼인 관계를 제외한 나머지 관계를 말하며, 君臣 관계, 上下 관계, 主從 관계, 師弟 관계, 同志 관계, 同僚 관계 등을 포함한다. 나방현은 역사 콘텐츠에 있어서는 사람의 역할이 다양하므로 보다 확장된 관계 모델을 필요로 한다고 하고, 사람과 사람의 관계로 가족 관계, 친구 관계, 가르치고 배우는 관계, 전문적 협업에서의 관계 등을 제시하고, 서비스 시나리오에 따라 중요도 또는 우선 순위를 결정하여 점차적으로 확대 구축하여야 할 것이라고 하면서, 그 예로 전쟁과 관련하여 적군, 아군, 공격·방어 등과 같은 관계어를 들었다.²⁵⁰⁾

본 논문에서는 사회적 관계로 친구, 스승, 제자 간의 관계를 'isFriendOf, hasDisciple(isDiscipleOf)'로 표현했으며, 묘지명을 작성할 때 撰·書·篆한 관계를 각각 'compose, writes, engrave'로 표현하였다. 撰, 篆, 書가 중요한 이유는 撰者는 비문의 대상자나 혹은 후손과 관직의 동료이거나 같은 문인, 친구 등 절친한 신분 관계를 유지하고 있다는 것 때문이다.²⁵¹⁾ 전통 시대 계급상 상하 관계를 표현하기 위해 'subordinate, superior'를 두었다. 또한 주종 간의 관계를 표현하기 위해서 'master, servant'를 두었다.

사회적 관계에서 신분 관계도 빼놓을 수 없다. 신분 관계는 역사를 이해하는데 중요한 단서를 제공한다. 아래 문장은 '王友珪의 僕夫인 馮廷諤이 朱全忠을 시해했다'는 내용으로 馮廷諤과 王友珪의 관계가 신분을 통해서 드러난다.

■ 梁郢王友珪 僕夫馮廷諤 弑梁主 友珪即帝位

☞ 梁郢王友珪 subordinate 馮廷諤

☞ 馮廷諤 hasStatus 僕夫

이 사건의 배경 지식을 살펴보면 馮廷諤이 後梁 太祖 朱全忠을 시해하고, 郢王 朱友珪가 황제에 즉위한다. 그런데 馮廷諤은 梁郢 王友珪의 僕夫이다. 僕夫는 마부 혹은 종을 말한다.

250) 나방현, 「U-City 정보서비스의 접근성 향상 방법에 관한 연구」, 한세대학교 대학원

박사학위논문, 2010, 47쪽.

251) 김우임, 「서울·경지지역의 朝鮮時代 士大夫 墓制 研究」, 고려대학교 대학원 박사학위논문, 2007, 53쪽.

여기서 郢王 朱友珪과 馮廷諤의 관계를 살펴보면 馮廷諤은 朱友珪의 마부이다. 『通鑑節要』에서는 朱友珪 직접적으로 朱全忠을 살해하도록 명령한 내용은 없지만, 전후 맥락을 통해서 朱友珪의 사주에 의해서 朱全忠을 살해한 것을 미루어 짐작할 수 있다. 그런데 이러한 신분 관계를 막상 프로퍼티로 표현하기에는 쉽지 않다. 사람과 사람의 관계는 Object Property인 ‘subordinate’를 사용하여 상하 관계를 연결하고, 한 사람의 신분은 신분 클래스와 Object Property인 hasStatus를 사용하여 연결해 준다.

```

<owl:NamedIndividual rdf:about="&knoPerson;郢王朱友珪">
  <rdf:type rdf:resource="&kno;Person"/>
  <kno:subordinate rdf:resource="&knoPerson;馮廷諤"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="&knoPerson;馮廷諤">
  <rdf:type rdf:resource="&kno;Person"/>
  <kno:status rdf:about="&kno;僕夫"/>
</owl:NamedIndividual>

```

아래의 ‘심복[腹心]’의 경우에도 상하 관계를 나타낸다. 상하 관계는 상사를 기준으로 ‘subordinate’로 관계를 표현한다.

- ☞ 蔣玄暉는 朱全忠의 심복이니,
- ☞ 蔣玄暉 - 심복이다 - 朱全忠

위의 경우에는 관계를 ‘심복이다’로 표현할 수도 있다. 그러나 이러한 관계를 모두 표현하려면 클래스의 설계가 계속해서 변할 수 있다. 기본 설계가 달라진다는 것은 기초가 흔들리는 것이기 때문에 가급적이면 설계 자체가 바뀌는 일은 최소화해야 한다. 그렇기 때문에 이런 관계는 포괄적인 개념인 ‘subordinate’로 표현한다.

사회적 관계는 이들 관계 이외에도 단순히 알고 지내는 관계는 knows로, 이들 관계에 포함이 되지 않는 관계를 표현하고자 할 경우에 isRelateTo로 표현하였다.

혈연 관계나 사회적 관계의 공통점은 정의역과 치역이 모두 Person이라는 점이다. 사회적 관계에 대해서 정리하면 다음과 같다.

표 45. hasRelatedSocial

속성		정의
hasRelatedSocial		社會的 關係
	isFriendOf	벗, 交友, 友
	hasDisciple	문인으로 두다
	isDiscipleOf	문인이 되다
	compose	묘지명을 撰한 관계
	engrave	묘지명을 篆한 관계
	subordinate	계급 상 아래 사람
	superior	계급 상 윗 사람
	master	주인
	servant	종
	writes	묘지명을 書한 관계
	knows	단순히 안면이 있는 관계
	isRelatedTo	위의 관계를 제외한 관계

2) 이벤트 시간 관계(eventTimeRelation)

앞서 역사적 사건을 단기적인 사건과 장기적인 사건으로 구분할 수 있다고 하였다. 장기적인 사건은 사건이 발생한 시점과 끝나는 시점이 존재한다. 이들 사건 간에는 서로 선후 관계나 포함 관계 등이 존재한다. 이벤트 시간 관계는 이러한 사건 간의 선후 관계를 표현한다. W3C에서 만든 Time Ontology in OWL Working Draft에 의하면 시간의 선후 관계를 역 관계를 포함해 모두 14개로 구분하였다. 이러한 분류는 기본적으로 James Allen이 정의한 13개의 규칙을 거의 그대로 도입한 것이다.²⁵²⁾

252) Allen, James F, George Ferguson, 「Actions and events in interval temporal logic」, 『Journal of logic and computation』, 4 5, 1994, 10쪽.

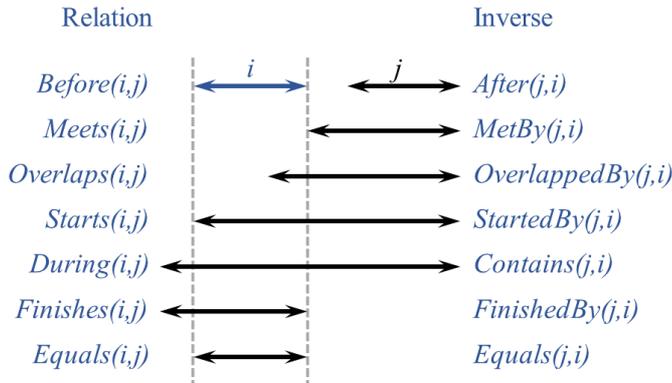


그림 37. 기간 간 발생할 수 있는 관계의 유형²⁵³⁾

위의 그림에서는 기간 간의 관계만을 표현했지만, 실제로는 기간이 아닌 시점도 동일하게 적용할 수 있다. 즉 시점과 시점, 시점과 기간, 기간과 기간 간에 모두 적용될 수 있는 관계인 것이다. 시간 관련해서는 시간의 선후 관계, 포함 관계 등을 다음과 같이 표현하였다.

표 46. eventTimeRelation

속성		내용
eventTimeRelation		사건 간 선후 관계
	after	후행 시간
	before	선행 시간
	contains	기간을 포함하다
	during	기간
	equals	동일하다
	finishes	동시에 끝내다
	finishedBy	동시에 끝나다
	meets	인접하다
	metBy	인접되다
	overlaps	겹치다
	overlappedBy	겹쳐지다
	starts	동시에 시작하다
	startedBy	동시에 시작되다

3) 의미역(eventRoleObject)

역사서에 나타나는 지식을 단순한 주어-술어-목적어의 RDF 트리플로 표현하는 것은 쉽지 않다. 앞서 말한 것처럼 ‘項羽가 義帝를 죽이다’라는 간단한 문장이라도 역사서의 특성상

253) W3C, Time Ontology in OWL, <https://www.w3.org/TR/2017/WD-owl-time-20170202/#AF-97>

시간을 기록해야 하며, 원인과 결과, 그리고 사건이 일어난 공간 등을 표현해 주어야 한다. 그러나 현실적으로 RDF 트리플의 이항 관계로는 관련된 지식을 표현하는 것이 쉽지 않다. 이를 해결하고자 공백 노드를 이용하는 방법과 중계자를 이용하는 방법이 연구되었다. 그러나 이러한 방법을 사용하더라도 모든 문장을 정보의 소실 없이 표현하는 것은 쉽지 않다. 그리하여 이를 해결하기 위한 방법으로 본 논문에서는 국어 문법에서 사용하는 意味役이라는 개념을 도입하였다.

어떤 동사나 형용사든지 그것이 한 문장을 완성하기 위해서는 필수적으로 요구하는 성분들이 있는데 이것을 필수 보어(complements)라고 부른다. 술어의 보어는 대부분 명사구(혹은 한정사구)로 실현되고, 언어에 따라 전치사구나 형용사구, 혹은 부사구로 실현될 수도 있다. 필수 보어들은 통사적으로 술어와 맺는 관계, 즉 문법적 관계(grammatical relation)에 따라 주어, 목적어, 혹은 간접 목적어 등으로 분류된다. 필수 보어는 술어와 통사적 관계를 맺을 뿐만 아니라 특정한 의미적 관계를 또한 맺게 되는데, 이것을 보어의 의미역(semantic role)이라고 부른다.²⁵⁴⁾ 만약 동사나 형용사를 중심으로 意味役에 따라서 적절하게 관계를 설정하면 복잡한 문장까지도 의미의 상당수 부분을 표현할 수 있다. 意味役은 동사에 따라 필요한 論項이²⁵⁵⁾ 달라진다. 예를 들어 ‘擊(공격하다)’라는 동사는 공격을 하는 행동주(agent)와 공격을 당하는 피동주(patient)가 있을 수 있다. ‘幸(가다01)’라는 동사는 이동하는 주체가 있으며, 중간 경로가 올 수 있고, 도착하는 목적지(goal)가 있어야 한다. 동사에 따라 꼭 필요한 구성 요소에 따라 의미를 표현할 수 있다.

■ 羽聞之烹韓生

☞ 項羽[agent]가 韓生[patient]을 삶아 죽이다.

■ 荊軻至咸陽

☞ 荊軻[agent]가 咸陽[goal]에 이른다.

여기서 동사는 彭과 至이다. 이 동사가 필요로 하는 필수 구성 요소는 項羽와 韓生, 烹嘉와 咸陽이 된다. 동사와 이들 논항과의 관계는 項羽와 烹嘉는 행동주가 되며, 韓生은 對象, 咸陽은 목적지가 된다. 여기서는 국어 문법에서 적용하는 意味役을 그대로 적용할 수 있는지에 대해서 살펴본다. 그러기 위해서는 먼저 실제로 적용하기에 앞서 역사적 지식을 어떤 방식으로 전달해야 하는지 고민해야 한다. 역사서에 나타나는 행위는 다분히 주관적일 수밖에

254) 남승호, 『한국어 술어의 사건 구조와 논항 구조』, 서울: 서울대학교출판부, 2007, 5쪽.

255) 김영희는 「논항의 판별 기준」에서 논항을 의미적 완결성을 지닌 완전문(complete sentence)을 구성하기 위하여 서술어 기능을 가지는 어휘가 자신의 자리값(valency)과 의미격 틀(θ-frame)에 따라 문장에 나타날 것을 요구하는 필수 성분이라고 하였다.

에 없다. 孔子는 『春秋』를 지을 적에 春秋筆法에 의해 사실을 간략히 기록했을 뿐, 비평이나 설명은 철저히 삼갔다고 한다. 褒貶의 원칙에 따라 용어를 철저히 구별하여 서술하였으니 사람이 죽었을 때도 대상이나 명분에 따라 ‘弑’와 ‘殺’을 구분하고, 다른 나라를 쳐들어갔을 때도 ‘侵’, ‘伐’, ‘入’, ‘取’ 등의 표현을 구분해 사용했다고 한다.²⁵⁶⁾ 그런데 이러한 표현의 차이에도 불구하고 사실상 전달하고자 하는 내용은 결국 ‘죽이다’, ‘죽다’로 수렴된다. 그렇다면 이러한 모든 느낌의 차이까지 그대로 전달해야 하는지를 고민해야 한다.

역사서에 나와 있는 ‘죽이다’와 ‘시해하다’라는 어휘는 ‘죽이다’는 의미를 포함하고 있음에는 변함이 없으나, 저자가 역사서를 집필할 때 집필자의 가치 판단 기준에 따라 적절한 어휘를 선택한 것이다. 그러므로 본 연구에서도 양측을 모두 수용하는 절충적인 방법을 취했다. 의미의 전달을 위해서 모든 문장을 그대로 변환하는 것이 아니라 중요한 내용을 전달하는 것으로 하였다. 또한 위의 예처럼 저자의 분명한 의도가 드러난 것은 가급적 사용된 어휘를 그대로 표현하였으며, 나머지 경우는 어휘의 표현에 크게 관계없이 정확한 의미 전달에 충실하고자 하였다.

다음으로 국어 문법에서의 의미역과 역사적 사실을 기술하기 위한 의미역의 차이를 살펴 보았다. 첫째 국어 문법에서 의미역을 부착하는 것은 전체 문장에 대해서 의미역으로 식별되는 요소를 빠짐없이 분석하나, 본 논문에서는 사건의 전달에 의미가 있는 요소만을 부착한다. 예를 들어 울산대학교 한국어처리연구실에서는 의미역 부착 말뭉치를 아래와 같은 형태로 구축한다. 모든 어휘는 『표준국어대사전』에서 부여한 구분 번호를 부착하고 동사에 따라 필요한 논항에 대해서 [ORG], [LOC] 등의 의미역 태그를 부착한다.

- 한국 관광객 16명 홍콩서 3중 추돌로 부상
- 한국_05[ORG]/NNP 관광객/NGG 16/SN+명_03/NNB+/,SP 홍콩[LOC]/NNP+서/JKB 3/SN+중_04/NGG 추돌/NGG로+사고_12/NGG+로/JKB 부상_05/NGG

그러나 본 논문에서는 의미 전달에 필요한 요소만을 대상으로 관계를 표현하고자 했다.

둘째, 국어 문법에서 사용하는 의미역을 도입하여 역사적 지식을 RDF 트리플로 표현할 때 국어학에서 문장을 문법적으로 분석하는 것과 달리 엄밀한 문법적 적용보다는 정확한 내용의 전달과 논리적 추론의 유효성을 우선으로 한다. 그 이유는 유사한 의미를 지니는 의미역이 동사나 조사에 따라서 각각 의미역을 달리하기 때문이다. 이러한 국어학의 논항을 그대로 적용시킬 경우에 논리적 추론이 어려울 수 있다. 예를 들자면, 아래의 경우 기존 문

256) 동양고전해제사전, 전통문화연구원,

http://db.cyberseodang.or.kr/front/alphaList/BookContent.do?mId=m02&bnCode=jti_5a1001&titleId=C50

법에 의하면 대상을 의미하는 ‘-을/를’과 결합하는 ‘齊王’은 대상(theme)으로 분류해야 한다. 그러나 본 논문에서는 추격이라는 어떤 힘이 미치는 개체로 보고 피동주로 분류하였다.

- 韓信이 동쪽으로 齊王을 추격하니

그렇기 때문에 조사에 따라서 意味役이 달라지는 것에 대해서 내용의 전달 차원에서 문법과 관계없이 의미 전달과 추론에 적합한 意味役을 선택하며, 여러 가지 사용되는 용어도 의미 전달에 큰 무리가 없다면 가급적 통일된 언어로 전달하고자 했다.

본 논문에서 역사적 지식을 RDF 트리플로 표현하기 위해 도입한 意味役은 연구자에 따라서 약간씩 다르다. 이러한 의미역 구분의 차이에 대해서 정리하면 아래와 같다. 이 중에서 역사적 지식의 표현에 필요한 의미역만을 사용하는 것으로 하였다.

표 47. 의미역(Semantic Role) 비교

세종전자사전 ²⁵⁷⁾	남승호 ²⁵⁸⁾	박철우·김종명 ²⁵⁹⁾	UCorpus ²⁶⁰⁾
행위주(AGT)	행동주(agent)	행위주(Agent)	행동주(AGT)
경험주(EXP)	경험주(experiencer)	경험주(Experience)	경험주(EXP)
영향주(EFF)	원인(cause)	영향주(Effector)	원인(CAU)
심리행위주(MAG)		심리행위주(Mental Agent)	
	피동주(patient)		피동주(PAT)
동반주(COM)	동반자(companion)	동반주(Companion)	동반자(COM)
대상(THM)	대상(theme)	대상(Theme)	대상(THM)
장소(LOC)	처소(location)	장소(Location)	처소(LOC)
출발점(SRC)	기점(source)	출발점(Source)	기점(SRC)
	경로(route)		경로(ROU)
도착점(GOL)	착점(goal)	도착점(Goal)	착점(GOL)
방향(DIR)	방향(direction)	방향(Direction)	방향(DIR)
도구(INS)	도구(instrument)	도구(Instrument)	도구(INS)
	자극(stimulus)		자극(STM)
	자격(qualification)		자격(ROL)
기준치(CRT)	비교기준(criterion)	기준치(Criterion)	비교기준(CRT)
			시간(TMP)
			정도(DGR)
			방법(MNR)
			재료(MAT)
			수혜자(BEN)
결과상태(FNS)		결과상태(Final State)	
목적(PUR)		목적(Purpose)	목적(PUR)

내용(CNT)		내용(Content)	내용(CNT)
15개	15개	15개	22개

가) 행위(action)

행위는 意味役에서 행동주가 하는 구체적인 행위로, 주로 동사나 형용사로 나타난다. 앞서 역사적 사건을 기술하기 위해서는 이항 관계로 표현하는 것이 한계가 있기 때문에 다항 관계를 표현하기 위해서 意味役을 도입하였다고 하였다. 意味役이라는 것은 결국 동사나 형용사가 필요로 하는 논항을 의미한다. 그러나 RDF 트리플의 주어-술어-목적어의 단순 구조로는 이를 직접 표현할 수 없으므로 이벤트를 중심으로 나머지 논항과의 관계를 표현해야 한다. 이때 이벤트와 행위를 연결해 주는 프로퍼티가 바로 action이다.

- 羽聞[agent]之 烹[action]韓生[patient]
 - ☞ 項羽[agent]가 韓生[patient]을 삶아 죽이다[action].
- 荊軻[agent]至[action]咸陽[patient]
 - ☞ 荊軻[agent]가 咸陽[goal]에 이르다[action].

이는 개체와 개체 간의 발생한 어떤 행위를 직접 표현할 수 없으며, 이벤트라는 노드를 통해서 이벤트와 행위를 연결하는 방식으로 표현해야 한다는 것을 의미한다. 이러한 표현은 개체와 개체를 직접 연결하는 것에 비해 논리적인 추론이 복잡해질 수밖에 없다. 이러한 문제는 앞으로 해결해야 할 과제이다.

행위와 상태를 표현하기 위해 동사나 형용사를 Action 클래스의 인스턴스로 표현한다는 것은 언어와 크게 관계 없이 자유롭게 행위나 상태를 표현할 수 있다는 의미가 된다. 그렇기 때문에 이들 행위나 상태를 나타내는 동사 혹은 형용사의 분류 체계를 잘 세워 적절한 의미를 확장하거나 연계시키는 것이 무엇보다 중요하다.

한문은 기본적으로 우리말 어순과 다르고, 또한 명사와 동사의 구분이 명확하지 않은 경우도 적지 않다. 행위 명사와 사건 명사에 ‘하다’가 붙으면 동사로써 자격을 가지는 경우가 많다. 예를 들어 ‘공부하다’라는²⁶¹⁾ 동사는 ‘工夫’라는 명사에 동사 파생 접미사인 ‘하’가 붙

257) 국립국어원, 『21세기 세종계획 전자사전 개발』, 2007, 59쪽.

258) 남승호, 앞의 책(2007), 47쪽.

259) 박철우·김중명, 『한국어 용언 사전 기술을 위한 의미역 설정의 기본 문제들』, 『語學研究』, 제41권, 제3호, 2005, 546~547쪽.

260) 김윤정·옥철영, 『한국어 서술어와 논항들 사이의 의미역』, 2014, 148쪽.

261) ‘공부하다’는 형태소 분석 시 ‘공부__01/NNG+하/XSV+다/EC’와 ‘공부하다’ 두 가지 경우를 모두 인정한다.

어서 하나의 동사로써 자격을 가지게 된다. 漢文에서 ‘建國’은 우리말로 ‘나라를 세우다’라고 번역할 수도 있지만 ‘건국하다’처럼 ‘建國’이라는 명사와 동사 파생 접미사와 종결어미를 결합한 ‘하/XSV+다/EF’를 사용하여 동사로 번역할 수도 있다. 南은 남쪽이라는 방향을 나타내기도 하지만 ‘南進하다, 南下하다’라는 동사로 사용되기도 한다. ‘怒’는 ‘노하다’라는 별도의 論項이 필요없는 경우도 있고, 어떤 구체적인 대상에 대해 ‘노여워하다’라는 의미로 번역되기도 한다. 그렇기 때문에 관계의 표현은 한자어를 사용할 수도 있다.

그리고 동사와 형용사로 표현되는 Action 클래스의 인스턴스는 계층 구조를 지닌다는 것은 계층 구조에 워드넷을 활용할 수 있다는 것을 말한다. 그러나 현재까지 한자어의 워드넷은 찾아 볼 수 없으며, 국어를 포괄적으로 다루는 워드넷은 코어넷이나 울산대학교의 UWordMap 등의 몇 개가 존재한다. 울산대에서는 2002년부터 『표준국어대사전』의 어휘를 기반으로 다의어 수준의 어휘의미망을 구축하기 시작하여, 서술어의 필수 논항이 가지는 의미제약정보를 명사어휘망의 최소상계노드로 설정하고, 부사와 의미적으로 연결될 수 있는 용언, 부사, 명사 등을 연결하여 명사, 용언, 부사 어휘들이 상호유기적으로 연결시켰다.²⁶²⁾ 그런데 이들 워드넷은 대부분 현대어에 특화되어 있어 고전을 번역한 번역서에 나타나는 수많은 어휘를 수용하기에는 아직 부족한 점이 많다.

나) 주어(subject)

(1) 행동주(agent)

국어 논항에서 어떤 행위나 상태를 일으키거나, 인식하는 주어를 행동주(agent)라 말한다. 행동주는 의도성을 가지고 행위를 발생시키는 주어뿐만 아니라, 어떤 상태를 인지하거나 지각하는 주어까지 포함한다. 행동주는 사람 이외에도 기관, 단체, 기타 유정물도 될 수 있다. 국어 문법에서는 경험주(experience)를 별도로 분류하기도 하나, 주어의 위치에 있는 요소는 행동주와 다음에 설명하는 영향주로만 구분한다. 경험주는 의도를 가지고 어떤 행위를 일으키는 것이 아니라 어떤 행위나 상태를 인식하는 개체에 부여되는 의미역이다. 이는 ‘불안하다, 좋다’ 등의 인지(cognition), 지각(perception), 감정(emotion)의 주체가 되는 논항의 의미역을 말하는데, 서술어가 심리를 표현하는 술어일 경우 의미역으로 경험주를 취하게 된다.²⁶³⁾ 국어 문법에서 심리술어라는 범위가 정확하게 구분되어 있지 않으며, 경험주도 대부분 사람인 점을 고려하였으며, 주어, 술어, 목적어로 표현하는 RDF 트리플의 특성을 감안하여 이러한 경험주도 행동주의 범주 안에 포함시켰다.

262) 배영준·옥철영, 「한국어 어휘지도(UWordMap)와 API 소개」, 『한국정보과학회언어공학연구회

2014년도 제26회 한글 및 한국어 정보처리 학술대회』, 2014년, 27쪽.

263) 김윤정·옥철영, 「한국어 서술어와 논항들 사이의 의미역」, 2014, 144쪽.

- 全忠[agent]既克蔡州[theme]兵勢益盛
 - ☞ 朱全忠[agent]이 蔡州[theme]를 점령하니, 군세가 더욱 강성하였다.
- 唐主[agent]祀南郊[location] 大赦
 - ☞ 唐主[agent]가 南郊[location]에 제사 지내고 크게 사면령을 내렸는데,

(2) 영향주(effector)

영향주(effector)는 동사가 나타내는 사건이나 형용사가 나타내는 현상을 비의도적으로 유발하는 주어를 말한다. 보통 無情物 주어인 경우가 많으며 역사서에서는 日蝕, 月蝕 등의 천문 현상, 지진 등의 자연 현상 등이 여기에 해당된다. 영향주는 도구와 거의 유사하나 영향주는 따로 행동주가 나타나지 않지만 도구는 조사 ‘-로, -으로, -에’와 함께 실현되면서 행동주가 함께 나타나는 구성을 보인다.²⁶⁴⁾

- 糧盡矢[effector]竭하여 城[effector]陷
 - ☞ 화살[effector]이 떨어져 성[effector]이 함락되니,
- 日食[effector]
 - ☞ 日食[effector]이 있자,
- 地震[effector]
 - ☞ 地震[effector]이 일어났다.
- 軍勢[effector]大振
 - ☞ 군세가 크게 떨쳐졌다
- 石乞[agent]이 以戈擊之
 - ☞ 石乞[agent]이 子路[patient]를 창[instrument]으로 공격하여 갓끈을 끊었다.

(3) 동반주(companion)

동반주(companion)는 독립적으로 역할을 하지 못하고 다른 의미역, 즉 행동주나 대상을 보조하여 그것과 같은 역할을 하는 개체에 부여되는 의미역으로 공동격(commutative) 표지 ‘-와/과’와 함께 나온다. 이러한 동반주의 판별은 매우 단순하다. 아래처럼 조사 ‘-와/과’와 결합되는 명사구는 동반주의 의미역을 가지는 것으로 본다.²⁶⁵⁾ 그러나 대상을 단순히 나열

264) 박철우·김종명, 위의 논문(2005), 562쪽.

265) 박철우·김종명, 앞의 논문(2005), 553쪽.

하는 것은 동반주가 아니므로 주의해야 한다.

- 智伯[agent] 怒 帥韓[companion]魏之甲 以攻趙氏
☞ 지금 智伯[agent]이 韓氏[companion]와 魏氏의 군대를 거느리고
- 二子[agent]乃陰與張孟談約[companion]
☞ 二子[agent]가 마침내 은밀히 張孟談[companion]과 약속하고는
- 文侯[agent]以卜子夏[patient], 田子方[patient]으로 爲師
☞ 文侯[agent]가 卜子夏[patient]와 田子方[patient]을 스승으로 삼고

위의 문장에서 ‘卜子夏[patient]와 田子方[patient]을’이 피동주가 되는 것은 ‘卜子夏와 田子方’이 접속조사 ‘와’로 연결이 되었지만 文侯와의 관계를 보면 실제로는 ‘文侯가 卜子夏를 스승으로 삼다’와 ‘文侯와 田子方을 스승으로 삼다’라는 두 개의 문장이 하나로 결합된 형태이다. 이 때문에 피동주가 되는 것이다. 그러나 아래와 같이 하나의 명사구 속에 ‘-와/-과’로 결합되어 있는 경우에는 그 전체가 하나의 명사구로서 역할을 하는 것으로 보기 때문에 ‘韓愈’와 ‘陸贄’, ‘趙高’와 ‘楊素’, ‘惠帝’와 ‘懷帝’는 하나의 명사구로서 행동주로 보고, 동반주로 보지 않는다.

- ☞ 韓愈[agent]와 陸贄[agent]는 글을 올려
- ☞ 趙高와 楊素의 간악함으로 扶蘇[patient]와 楊勇[patient]을 죽게 만들었으니,
- ☞ 惠帝[agent]와 懷帝[agent]가 친애함이 없어서

(4) 피동주(patient)

피동주(patient)는 기본적으로 행동주의 행위에 영향을 받아 변화를 겪는 논항이다. 보통 피동문에 나타나는 주어는 피동주로 보는 데 비해, 본 논문에서는 피동주의 범위를 확대하였다. ‘피동주(patient)’는 상태 변화를 겪으며, ‘대상(theme)’은 처소의 변화를 겪는다. ‘피동주’는 그 행위에 의해 어떤 변화를 겪는 데 비해 ‘대상’은 이동이나 위치가 변화된 뿐 그 개체 자체는 아무런 영향을 받지 않는다.²⁶⁶⁾

- 政事를 一委令孜[patient]
☞ 정사를 한결같이 田令孜[patient]에게 위임하였다
- 削田令孜[patient]官爵 長流端州

266) 남승호, 앞의 책(2007), 33쪽.

- ☞ 田令孜[patient]의 관작을 삭탈하여 멀리 端州로 유배보냈다.
- 令狐潮圍張巡[patient]於雍丘
 - ☞ 令狐潮가 張巡[patient]을 雍丘에서 포위하여
- 鎮西北庭行營兵이 亦殺節度使荔非元禮[patient]
 - ☞ 鎮西北庭의 行營 병사들이[agent] 또한 節度使인 荔非元禮[patient]를 죽이고
- 豫讓 伏於橋下 襄子至橋 馬驚 索之 得豫讓[patient] 遂殺之
 - ☞ 襄자가 다리에 이르자 말이 놀라므로 수색하여 豫讓[patient]을 잡아서 마침내 죽였다

일반적으로 국어 문법에서 아래의 ‘축구공’과 ‘범인’을 피동주로 보고 있다. ‘붙잡혔다’라는 피동문에 나타나는 주어는 사건에 주체적으로 참여하는 행동주가 아니기 때문에 사건에서 영향을 입는 피동주로 구분한다.

- 연이[agent]가 축구공[patient]을 힘껏 찼다.
- 범인[patient]이 지난 밤 시내 호텔에서 붙잡혔다.

그러나 본 논문에서는 피동과 사동을 구분하지 않고 동사(사동, 피동)를 동일한 범주의 행위로 보고, 동사의 주어를 행동주로, 그 행위의 대상이나 영향이 미치는 것을 피동주로 보았다. 위의 문장에서는 ‘붙잡혔다’라는 행위의 주어는 범인이 되며, 동사의 목적어로 구현되는 것을 피동주로 보며, 피동주가 사람이 아닌 경우에는 대상으로 보아, 아래와 같이 분석하였다. 이는 RDF 트리플에서는 조희의 편의성을 위해 가급적이면 문법적인 요소를 단순화시킬 필요가 있기 때문이다.

- 연이[agent]가 축구공[theme]을 힘껏 찼다.
- 범인[agent]이 지난 밤 시내 호텔에서 붙잡혔다.

(5) 기타주어(etcsubject)

위의 예 이외에 주어이면서도 클래스로 분류하기 어려운 기타 유형을 기타주어(etcsubject)로 표현한다.

다) 대상(theme)

국어 문법에서 대상은 서술어의 행위에 의해 옮겨지거나 묘사되는 등의 행위나 과정의 영향을 받는 논항을 의미한다. 행동주와 같이 의도적으로 무엇인가를 할 수는 없지만, 그

행위나 과정에서 없어서는 안 되는 필수 논항에 해당하는 것으로 보고 있다.²⁶⁷⁾ 본 논문에서는 이러한 구분을 단순화시켜 행위의 영향을 받은 사람을 피동주로 보고, 나머지 지명이나 사물 등은 목적어를 의미하는 것을 보고 대상으로 표현한다.

- 史思明[agent] 引兵[theme]攻河陽[theme]
 - ☞ 史思明[agent]이 군대를[theme] 이끌고 河陽[theme]을 공격하였다.
- 帝[agent]善之 詔相[theme]給事中[goal] 皆從其議
 - ☞ 황제[agent]가 이를 좋게 여기 魏相[patient]을 給事中[goal]에 임명하고 모두 그의 의논을 따랐다.

라) 처소(location)

처소(location)는 어떤 일이나 사건이 발생한 위치, 장소, 공간 혹은 사물의 위치 등을 나타내는 의미역으로 대체로 조사 ‘-에, -에서’로 표시된다. 처소 의미역은 실제 장소의 의미를 가진 어휘들뿐만 아니라 신체의 일부를 나타내기도 한다.²⁶⁸⁾ 또한 같은 공간이 처소나 도착점을 지칭하기도 한다. 이때 도착점은 이동을 하는 행위가 어떤 목적지를 가지는 데 비해, 처소는 특별한 목적지를 향해서 이동하는 것이 아닌 단순히 어떤 행위가 발생한 공간을 의미한다.

- 卽皇帝位[goal]於天興殿[location]
 - ☞ 天興殿[location]에서 황제[goal]에 즉위하게 하니,
- 上皇[agent]이 卽日에 幸興慶宮[goal]하여 遂居之어늘
 - ☞ 上皇[agent]이 당일 興慶宮[goal]으로 가서 마침내 거처하였는데,

마) 기점(source)

기점(source)은 주로 ‘출발하다, 도착하다’ 등의 이동을 나타내는 동사나 ‘마꾸었다’ 등의 변화를 나타내는 동사에 결합한다. 이는 물리적 혹은 추상적이든 그 출발 지점 혹은 시작 지점을 의미하며, 주로 ‘-에서’를 ‘-로부터’ 등과 결합한다.²⁶⁹⁾ 이동과 관련한 의미역은 행위의 시발점을 가리키는 기점(source), 이동의 행위가 거쳐 가는 중간 경로(route), 그리고 이동의 방향(direction), 행위의 목적 지점을 가리키는 착점(goal) 등으로 분류한다.

267) 김윤정·옥철영, 앞의 논문(2014), 144쪽.

268) 박철우·김종명, 앞의 논문(2005), 556쪽.

269) 김윤정·옥철영, 앞의 논문(2014), 145쪽.

- 郭子儀[agent]等 將兵五萬 自河北[source] 至靈武[goal]
- ☞ 郭子儀 등이 5만의 병력을 거느리고 河北[source]으로부터 靈武[goal]에 이르니,

바) 경로(route)

경로(route)는 장소 표현으로 이루어진 논항이 이동의 행위가 거쳐 가는 중간 경로를 나타내는 개체에 부여하는 의미역을 말한다.²⁷⁰⁾ 보통 ‘~을 통해서’, ‘~로 해서’와 결합한다.

- 自閔河[source]로 沿潁[route]入淮[goal]
- ☞ 閔河[source]로부터 潁水[route]를 따라 淮水[goal]로 들어가게 하니
- 郭威[agent]自迎春門[route]入
- ☞ 郭威[agent]가 迎春門[route]으로 들어와

사) 목표(goal)

목표(goal)은 동사가 표상하는 사건이 물리적 이동을 포함하고 있을 경우 그 끝점을, 추상적인 행위나 태도의 의미를 포함하고 있을 경우 그 지향점을 나타내는 개체에 부여되는 의미역이다.²⁷¹⁾ ‘가다, 도망하다, 출발하다, 나아가다’ 등의 이동 동사의 경우 행위의 도착 지점을 가리키며, ‘임명하다, 고용하다, 삼다, 만들다, 나누다, 추대하다’ 등의 동사에서는 상태나 신분의 변화를 나타내기도 한다. 또한 추상적인 행위나 태도를 나타낼 때 그 지향점을 나타내기도 한다.

- 樂毅[agent]遂奔(奔)趙[goal]
- ☞ 樂毅[agent]가 趙나라[goal]로 도망가니,
- 翰林學士李吉甫[patient] 爲中書侍郎[goal]
- ☞ 翰林學士 李吉甫[patient]를 中書侍郎[goal]으로 삼아
- 帝[agent]善之 詔相[theme]給事中[goal] 皆從其議
- ☞ 황제[agent]가 이를 좋게 여겨 魏相[patient]을 給事中[goal]에 임명하고 모두 그의 의논을 따랐다.

‘樂毅’라는 行動主가 도망한 사건의 결과로 ‘樂毅’가 ‘趙나라’에 있게 됨을 의미한다. 處所 동사에서 倒着點에 goal을 사용하지만 관직 등을 임명할 때의 그 결과로 이루어진 상태의 결과도 goal로 표현한다. 위 문장은 ‘李吉甫’는 어떤 행위의 결과로 신분의 변화가 발생하여

270) 김윤정·옥철영, 앞의 논문(2014), 146쪽.

271) 박철우·김종명, 앞의 논문(2005), 558쪽.

‘中書侍郎’이 되었음을 표현한다. 상태의 변화 결과도 goal로 표현한다. 그런데 두 번째 문장에서 李吉甫를 中書侍郎으로 삼은 주체는 나타나 있지 않다. 한문에서는 이런 문장이 빈번하므로, 이럴 경우에 주어는 황제라고 추정할 수 있다.

아) 방향(direction)

방향은 말 그대로 이동의 方向을 가리키며 방향격 표지 ‘-로, -으로’와 함께 사용된다. 도착점과 유사하지만 도착점이라고 하기에는 끝점이 불분명하여 단순히 방향만을 나타낸다.²⁷²⁾ 방향은 사건 중에서 중요한 의미가 있을때만 부착하고 그렇지 않은 경우에는 별도로 표시하지 않는다.

- 因擁威[patient]南[direction]行이라
☞ 인하여 郭威[patient]를 옹위하고 남쪽[direction]으로 갔다.

자) 도구(instrument)

도구는 행위나 이동을 나타내는 동사와 결합하는 도구나 재료, 방편 그리고 어떤 목적을 이루기 위한 수단 등에 부여하는 의미역이다.²⁷³⁾ 이는 재료(material)를 별도로 구분하기도 하나 여기서는 별도로 구분하지 않았다.

- 使中郎將呂布로 持矛[instrument]刺卓하고
☞ 中郎將 呂布로 하여금 창[instrument]을 잡고 董卓을 찌르고
- 石乞[agent]이 以戈擊之
☞ 石乞[agent]이 子路[patient]를 창[instrument]으로 공격하여 갓끈을 끊었다.

차) 시간(time)

역사 자료의 특성상 시간 정보는 매우 중요한 의미를 지닌다. 특히 편년체 역사서는 원칙적으로 사건을 왕조, 연도순으로 기술하기 때문에 모든 사건의 기준은 연도에 의해서 전개된다. 또한 연도로 본문 중에 구체적인 시간을 기록하지 않았어도 전후 맥락에 의해 시간을 파악할 수 있는 경우가 대부분이다. 그러므로 연도를 특별히 관리할 필요가 있다. Year 클래스와 연결하기 위한 프로퍼티가 바로 time이다.

272) 김윤정·옥철영, 앞의 논문(2014), 146쪽.

273) 박철우·김종명, 앞의 논문(2005), 561쪽.

- 官軍이 出陰地關 不戰而走
 - ☞ 官軍이 陰地關을 나가 싸우지 않고 패주하였고

위의 내용은 당나라 황제가李克用의 관작을 삭탈하고, 張濬을 시켜서李克用을 정벌하게 했는데 관군이 싸우지 않고 달아났다는 내용이다. 이 사건은 890년 5월에 발생하였다. 편년체 사서이기 때문에 시작 부분에 연도를 표기하므로 그 연도를 참조하여 구체적인 시간을 표현해 주어야 한다.

카) 내용(content)

내용(content)은 ‘말하다’ 등의 발화 동사, 인지 동사, 평가 동사 등의 경우, 발화, 인지, 평가의 내용을 나타내는 의미역이다.²⁷⁴⁾ 주로 인용절이 여기에 해당한다. ‘시키다, 명하다’ 하는 사역 동사에 내용은 직접 content로 표현할 수도 있지만 내용을 분해하여서 표현할 수도 있다.

- 恭, 顯以爲必無憂[content]
 - ☞ 弘恭[agent]과 石顯[agent]은 굳이 근심할 것이 없다고[content] 말했는데,

타) 부가어(adjunct)

국어 문법의 통사 이론에서 술어가 필수적으로 요구하는 논항을 보어(complement)라고 하고, 수의적으로 취하는 논항을 부가어(adjunct)라 하여 구분한다.²⁷⁵⁾ 보통 수의적인 논항은 반드시 필요한 것이 아니나 문장에 따라서 중요한 의미를 지니는 경우가 있다. 본 논문에서 부가어는 중요한 요소로 간주하지 않으며 사건을 기술할 때 중요한 의미를 지니고 있다고 판단되는 경우에만 부착한다.

(1) 정도(degree)

정도(degree)는 한문에서 ‘皆, 多, 大, 盡, 差’ 등이 정도를 나타낸다. 이들 용어는 우리말 ‘몹시, 무척, 크게, 가장, 극히, 매우, 아주, 너무, 되게, 굉장히, 참, 대단히, 엄청, 대개, 대부분’ 등으로 다양하게 번역이 된다. 정도를 나타내는 부사나 명사가 주로 그 대상으로, 우리말 문법에서 정도부사가 그 범주에 속한다. 정도부사란 어떤 상태나 동작의 정도를 나타내는 부사를 말한다.²⁷⁶⁾ 일반적으로 이런 부사어는 의미 전달에서 크게 중요한 역할을 하지는

274) 박철우·김중명, 위의 논문(2005), 563쪽.

275) 남승호, 앞의 책(2007), 25쪽.

276) 임규홍, 「국어 정도 부사의 화용화」, 『언어과학연구』, 제24집, 2003, 284쪽.

않는다. 그런데 아래 문장에서처럼 정도를 나타내는 부사나 명사가 없을 경우에 그 의미가 정확하게 표현되지 않는 경우가 있다.

- 營中 大[degree]亂
 - ☞ 적의 진영 안이 크게[degree] 혼란하였다.
- 吳, 楚士卒 多[degree]飢死叛散 乃引去 亞夫出精兵
 - ☞ 吳·楚의 사졸들이 대부분[degree] 굶어 죽고

(2) 빈도(frequency)

한문에서 ‘屢, 累, 屢次, 累次, 歲, 數’이 빈도(frequency)에 해당된다. 우리말로는 ‘빈번하게, 자주, 여러 번, 항상, 자주, 가끔, 거의, 전혀’ 등으로 번역된다. 이는 자주 행해지는 정도를 나타내는 어휘로, 정도와 비슷하게 문장에서 빈도를 꼭 표현해야 하는 경우에 사용한다.

- 唐主兵屢[frequency]敗하니
 - ☞ 南唐의 군대가 여러 번[frequency] 패하자
- 累[frequency]讓不獲하니
 - ☞ 鄭縻가 여러 번[frequency] 사양하였으나

(3) 수량(quantity)

역사서의 본문 내에 수량을 나타내는 어휘를 RDF 트리플로 표현해야 하는 경우가 간혹 있다. 아래 문장의 경우 만 명이라는 병력의 수는 특별한 의미가 있기 때문에 ‘quantity’라는 의미역을 사용하여 수량을 표현한다.

- 朔方兵[theme]萬人[quantity]하야 與之하다
 - ☞ 朔方の 병력[theme] 만 명[quantity]을 나누어 그에게[patient] 주었다

위에서 설명한 意味役을 정리하면 다음과 같다.

표 48. eventRoleObject

연번	프로퍼티	약칭	명칭	내용
1	action	AC	행위	구체적인 행위를 표현함
2	agent	AG	행동주	행위를 한 주체
3	effector	EF	영향주	無情物 주어
4	companion	CO	동반주	행위를 할 때 참여한 행위자
5	patient	PA	피동주	사건에서 영향을 입는 것, 상태의 변화를 겪는 것
6	theme	TE	대상	이동체, 대상
7	location	LO	처소	발생위치, , 장소명사와 함께 처소를 나타냄
8	source	SR	기점	기점, 출발점
9	route	RT	경로	경로, 지나온 경로
10	goal	GO	목표	도착점, 처소의 변화, 상태나 자격의 변화
11	direction	DR	방향	이동의 방향
12	instrument	IN	도구	도구
13	time	TM	시간	시간과 연계
14	content	CT	내용	내용의 표현
15	etcsbjeect	ES	기타주어	위의 주어로 속하지 않으나 꼭 필요한 주어
16	frequency	FR	빈도	빈도의 표현
17	degree	DE	정도	정도의 표현
18	quantity	QU	수량	수량의 표현

4) 국적(nationality)

전통 시대의 왕조는 개국한 시간과 멸망한 시간이 존재하므로 연도를 파악하는 데 유용한 정보를 제공한다. 이러한 國籍을 표현하기 위한 속성이 nationality이다.

5) 지명 정보(hasPlace)

공간과 관련하여 출생지, 사망지, 거주지, 활동지 등 여러 가지 유형의 공간과 관련된 정보가 있다. 역사 연구에 있어 출생지와 사망지, 거주지 등의 정보는 중요한 의미를 지니고 있다. 예컨대 榜目에서 거주지 정보가 그러한 예가 될 것이다. 아래의 속성으로 사용한 어휘는 DBpedia에서 설계한 온톨로지(dbpedia_2016-04.owl)를 참조하였다. DBpedia에서 설계한 온톨로지 중 일반적인 메타데이터 요소는 역사서를 표현하는 데도 유용하므로 사람의 출생과 사망, 그리고 시간 등은 이 온톨로지를 참조하였다. 그리고 지명 정보와 공간 정보는 구분해 주었다. 전자 지도에 사용하는 經度, 緯度 등의 좌표 관련 정보는 대부분 단순 수자의 나열이기 때문에 Datatype Property로 표현한다. 아래의 Object Property는 정의역은 Person이며, 치역은 Place가 된다.

표 49. 공간 정보 관련 오브젝트 프로퍼티

속성		내용
hasPlace		위치 정보
	birthPlace	출생지
	clan	본관
	deathPlace	사망지
	livingPlace	거주지
	nativePlace	출신지
	occurPlace	활동지
	penalColony	유배지
	prisonPlace	수형지
	tombPlace	묘, 장지 위치

6) 신분(hasStatus)

전통 시대에 신분은 중요한 의미를 지니고 있다. 신분의 분류라는 것이 아직 학계에서 공식적으로 인정받는 분류가 없으며 본 논문에서 다루기도 쉽지 않은 분야이다. 신분 분류는 클래스로 분류하였으며, 이 클래스의 인스턴스로 구체적인 신분을 등록할 수 있다. 어떤 사람의 신분을 표현하기 위해서는 'hasStatus' 속성을 사용한다. 정의역은 Person이고 치역은 Status이다.

7) 포폄(appraise)

역사를 편찬할 때 사건을 논평하며 인물을褒貶하는 기준이 되는 원칙을 書法이라 한다. 이러한 포폄은 정통을 어떤 나라에 두느냐와 서술하는 인물의 신분이나 인물이 행한 행동에 따라 다르게 표현되는데, 죽음을 나타내는 말과 정벌을 나타내는 말에 자주 사용된다. 사람이 죽었을 때 죽음을 표현하는 어휘를 신분에 따라 '崩, 薨, 殂, 卒, 死' 등으로 다르게 쓰는 것이 그러한 예일 것이다.

표 50. 죽음을 나타내는 포뮬어

어휘	통감절요 전체	오대십국 시대
崩	63	5
薨	47	3
殂	41	11
卒	74	16
死	28	5
합계	253	40

위의 어휘는 유교적 전통 시대의 역사가의 인식을 엿볼 수 있다는 점에서 중요하며, 이러한 서법은 조선 시대 유학자에게 많은 영향을 끼쳤다. 이들 어휘를 사용한 몇 개의 예시를 들면 다음과 같다. 이들 어휘는 ObjectProperty인 `appraise`로 표현한다.

표 51. 죽음을 표현하는 어휘의 예시

연번	인물명	한자	신분	기본형	비고
1	智宣子	卒	제후	죽다	
2	周威烈王午	崩	왕	죽다	
3	魏文侯	薨	제후	죽다	
4	楚悼王	薨	왕	죽다	
5	齊康公	薨	제후	죽다	
6	周安王驕	崩	왕	죽다	
7	周烈王喜	崩	왕	죽다	
8	秦孝公	薨	제후	죽다	
9	周顯王扁	崩	왕	붕하다	
10	周赧王定	崩	왕	죽다	
11	張儀	卒	정승	죽다	
12	楚懷王	薨	왕	죽다	

8) 기타

가) 서명 정보(bookInformation)

역사서에는 도서와 관련된 정보가 빈번하게 나온다. 그런데 이 책에 대해서 도서관 수준으로 정보를 부여할 수는 없다. 도서나 문서, 작품에 관련된 온톨로지 모델로는 국립중앙도서관에서 발행한 LOD 모델²⁷⁷⁾, 더블린코아에서 제공하는 모델 등을 참조할 수 있다. 본 연구에서는 역사서를 읽는데 꼭 필요하다고 판단되는 정보만을 설계에 포함하였으며, 상세한 정보는 `owl:sameAs`나 `owl:seeAlso` 등의 기능을 이용하여 원 소스를 제공하는 사이트와 연

277) 데이터셋, 국립중앙도서관, <http://lod.nl.go.kr/home/about/dataset.jsp>

결시키는 것으로 하였다. 그럼에도 불구하고 서명 정보가 필요한 이유는 도서관에서 제공하는 정보 이외에도 실전되어 현존하지 않는 도서가 존재하므로, 이들을 표현할 필요가 있기 때문이다. 서명 정보는 더블린코아에서²⁷⁸⁾ 사용하는 어휘 셋 중 고서에 적합한 것을 수용하였다. 제목, 저자를 서명정보의 하위 프로퍼티로 둔다.

표 52. 서명 정보

속성		내용
bookInformation		서명정보
	dcterms:title	제목
	dcterms:creator	생성자

나) 인과 관계(causalRelationship)

역사적 사건에서 인과 관계는 역사를 이해하는 데 있어 매우 중요한 역할을 한다. 인과 관계는 주어와 목적어 사이에서 발생하는 관계로, 정방향과 역방향 모두 표현할 수 있도록 'beCausedBy, cause'을 두었다.²⁷⁹⁾ 이들 관계는 역 관계(inverseOf)로 표현한다.

표 53. 인과 관계

속성		내용
causalRelationship		인과 관계를 표현
	beCausedBy	~에서 유래하다
	cause	~결과를 가져오다

기타 관력 정보나 사건의 선후 관계의 표현, 어휘 간 관계를 표현하는 관계 등의 프로퍼티를 가지도록 설계하였다.

다) 사건 관계

사건과 관련하여는 선행사건과 후행사건을 구분해 주었으며 사건 간 상하위 관계를 연결해 줄 수 있도록 하였다. 정의역과 치역은 모두 Event이다.

278) 한국 더블린코아 DCMI 공인사이트, <http://www.dublincore.go.kr/metadata-basics/term>

279) 유로워드넷에서는 원인과 결과를 CAUSES와 IS_CAUSED_BY로 표현하였다.

표 54. 사건 관계

속성		내용
hasEvent		인과 관계를 표현
	hasPostEvent	후행 사건을 가지다
	hasPreEvent	선행 사건을 가지다
	hasSubEvent	하위 사건을 가지다
	hasSuperEvent	상위 사건을 가지다

다음은 위에서 설명하지 않은 나머지 프로퍼티이다.

표 55. 기타 오브젝트 프로퍼티

속성		내용
hasCareer		경력 정보, 이력 정보
hasEraName		연호를 가지다
hasFamily		가문을 가지다.
hasPosition		관직을 가지다
hasRelate		연관 관계를 표현
hasRelatedOrganization		조직과 관련이 있다.
hasReligion		종교를 가지다
hasSource		출처로 하다
hasStatus		신분을 가지다
hasworks		저술, 저작, 작품을 쓰다
nationality		국적
receiveAward		공훈, 상을 받다

다. 데이터 타입 프로퍼티(DatatypeProperty)

1) 의미역(eventRoleData)

의미역은 event를 주어로 하여 목적어와의 관계를 표현하는 프로퍼티이다. 그런데 OWL에서 주어 위치에는 자원이 올 수 있고 목적어 위치에는 자원과 리터럴(값)이 올 수 있다. 이때 자원은 URI 참조를 이용하여 식별하며, Object Property로 관계를 연결해 주어야 한다. 반면에 리터럴은 Datatype Property를 이용하여 연결을 해야 한다. 그런데 의미역으로 연결해야 할 대상이 반드시 URI로 식별할 수 있는 것이 아닌 경우가 있다. 의미역 속성으로 연결해야 하는 대상이 어떤 경우에는 URI 참조를 사용하고, 어떤 경우에는 리터럴을 사용해야 하는 경우도 있다. 예를 들어 屢, 累, 屢次, 累次 등의 빈도나 萬餘騎, 萬餘人, 萬七千

人 등의 수량은 인스턴스로 관리하는 것보다는 리터럴로 처리하는 것이 낫다. 이런 이유로 목적어의 성격에 따라 Object Property는 프로퍼티 이름을 eventRoleObject로, Datatype Property는 eventRoleData로 정의하였다.

표 56. eventRoleData

연번	프로퍼티	내용
1	complement	기타 보어
2	frequency	빈도
3	quantity	양
4	distance	전, 후, 좌, 우 상하 Location의 주변을 나타내는 경우

다음으로 어휘 선택의 문제이다. 아래 문장에서 보면 ‘兵’은 우리말로 ‘전쟁’으로 번역되었는데, 그런데 兵은 전쟁뿐만 아니라 ‘兵士, 兵卒, 軍士, 軍人, 武器, 兵器’ 등 다양한 뜻을 가지고 있다. 『漢語大詞典』에서는 兵자에 ‘1.兵器 2.兵卒, 軍隊 3.軍事, 戰爭 4.用兵器殺人, 5.猶傷害, 6.謂戰死’ 등 6개의 뜻으로 분류하였으며²⁸⁰⁾, 『漢韓大辭典』에서는 ‘1.병기, 무기, 2.병졸, 군졸, 군대, 3.군사. 군대나 전쟁에 관한 일, 4.싸움, 전쟁, 5.무기를 사용하여 다 죽이다, 6.재앙, 재해, 7.막다, 방어하다, 8.전사’ 등 8가지 뜻으로 구분하였다.²⁸¹⁾ 정확한 의미의 전달을 위해서는 이들 한자가 가지는 의미를 모두 식별하고 식별된 의미를 사용해야 하나, 현실적으로는 시간과 예산의 문제로 그렇게 하기는 사실상 어렵다. 현실적인 대안은 개체를 정리하면서 가급적 원문에 사용된 어휘를 살려주면서도 그 문장에 적합한 어휘로 바꿔주는 방법이 있다. 예를 들어 아래 문장에서는 번역에서 사용한 戰爭이라는 개념으로 치환하는 것이다.

- 仍歲輸金帛百萬하여 以求罷兵
- ☞ 이어서 해마다 황금과 비단 백만을 바쳐 전쟁을 중지할 것을 청하였다.

2) 날짜(date)

날짜는 시간의 일종으로 연·월·일로 구성되어 있다. date 프로퍼티의 정의역은 Person이나 event가 된다. 날짜와 관련하여 먼저 year, month, day를 각각 분리해야 할지 아니면 통합하여 하나로 표현해야 할지를 검토해 보아야 하며, 음양력, 윤달 표기를 어떻게 할지에 대해서 검토해 보아야 한다. 날짜의 표현 문제는 RDBMS에서는 시스템의 성능을 좌우하는

280) 漢語大詞典 編寫委員會, 『漢語大詞典』2, 2008, 92쪽.

281) 단국대학교동양학연구소, 『漢韓大辭典』2, 단국대학교, 2000, 246쪽.

매우 민감한 문제이어서 이에 대한 충분한 검토가 필요한 사안이다. 그 이유는 대용량의 DB에서는 인덱스 사용 여부가 성능을 좌우하기 때문에 인덱스를 사용할 수 있는 형식으로 날짜를 표현해야 하기 때문이다. 그러나 LOD에 있어서는 상대적으로 날짜의 표현은 중요하지 않은 것으로 판단된다.

먼저 연·월·일의 분할 여부는 시간의 계산이 필요한지 여부를 검토해야 한다. ‘後周의 世宗 柴榮은 921년 10월 27일에 태어나서, 太祖 郭威의 양자가 되어 954년 2월 26일에 太祖에 이어 즉위하여 959년 7월 27일에 37세의 나이로 세상을 떠난다.’²⁸²⁾라는 문장으로 예를 들어 보겠다. 여기서 만약 世宗이 재위한 날짜 수가 얼마나 될까라는 질문이 필요하다면, 이에 걸맞는 데이터 설계와 유형의 선택이 필요하다. 이는 은행이나 통신사라면 날짜에 따라 이자가 달라지기 때문에 이를 정확하게 표현해야 하나 역사자료에서는 그 정도의 정밀성을 요구하지 않는다.

다음으로 음력과 양력의 구분과 윤달의 표현 여부를 결정해야 한다. 『通鑑節要』의 서비스만을 생각한다면 모두 음력을 사용했기 때문에 특별히 고민할 일은 없으나 역사 전반으로 확대한다면 구분해 줄 필요가 있다. 이와 관련하여 정도현 등은 연도를 기원전과 기원후를 구분하고, 기원전은 ‘-YYYYMMDD’으로, 기원후는 ‘YYYYMMDD’로 표현할 것을 제안하였다. 이때 윤달일 경우에는 맨 뒤에 ‘YYYYMMDD윤’이라고 표현하고, 연월일 미상일 경우에는 00으로 채운다.²⁸³⁾ 시간과 관련하여 더블린 코어(Dublin Core)에서는 date를 XMLSchema에서 규정하고 있는 ‘http://www.w3.org/2001/XMLSchema#date’ 형식을 사용하였다.²⁸⁴⁾ 한국사 LOD에서 시간과 관련된 프로퍼티는 ‘hasAge, hasDynasty, hasEndAge, hasEndDate, hasEndDynasty, birthDate, ComingEndDate, DeathDate’ 등인데, 이들은 모두 문자열인 ‘http://www.w3.org/2001/XMLSchema#string’으로 정의하였다. 날짜와 관련하여 XMLSchema에서는 ‘YYYY-MM-DDThh:mm:ss’처럼 더 엄밀한 날짜 표현 형식이 많이 있으나, 역사 자료에서는 그 정도의 수준까지는 필요치 않을 것 같다.

본 연구에서 날짜는 ‘beginDate(시작일), endDate(종료일), birthDate(출생일), deathDate(사망일)’ 등으로 구분하였으며 리터널 형식을 Literal으로 정의한다. ‘beginDate’와 ‘endDate’는 사건의 시작과 끝나는 지점을 표시하기 위한 프로퍼티이다.

282) Baidu, <https://baike.baidu.com/item/柴榮>

283) 김윤정·옥철영, 『한국어 서술어와 논항들 사이의 의미역』, 2014, 147쪽.

284) 예를 들어 ‘<dcterms:modified

rd:datatype="http://www.w3.org/2001/XMLSchema#date">2012-06-14</dcterms:modified>’처럼 표현한다.

표 57. date 프로퍼티

속성		내용
date		날짜
	startDate	시작일(사건, 수행, 종교, 관직 등)
	birthDate	생년
	deathDate	몰년
	endDate	종료일(사건, 수행, 종교, 관직 등)
	governmentComingDate	출사일
	honorDate	수상일
	created	생산일

3) 인명 정보(personInformation)

역사서에는 같은 사람이 여러 가지 이름으로 표현된다. 인물의 字, 號, 諡號, 廟號, 封號, 殿號, 尊號 등이 그러한 것들인데 이들 이름은 서로 같은 사람을 지칭하는 것임을 표시해 주어야 한다. 이름과 관련해서는 이름을 클래스로 표현해야하는 것이 좋을지 아니면 리터럴로 표현해야 할지를 결정해야 한다. 클래스로 표현한다는 것은 URI 참조를 사용하며, Object Property를 사용하는 것이며, 리터럴로 표현한다는 것은 Datatype Property로 표현한다는 것을 말한다. 본 논문에서는 인명의 이칭, 별칭 등은 Datatype Property로 표현하였다.

한국사 LOD에서는 Person 클래스를 두고 리터럴과 관계를 표현하기 위한 'DatatypeProperty'를 'aliasName, bonghoName, changedName, childhoodName, choName, jaName, myohoName, nickName, penName, posthumousName, realName, religionName' 등으로 두었다.

인명정보 클래스에서는 하위 클래스로 'personName, gender, legitimacy, relationshipOrder, personId, UCI'를 둔다. 'personName'은 인명의 다양한 명칭과의 관계를 표현하기 위한 프로퍼티이다.

Person 클래스에는 사람의 대표명칭을 인스턴스로 정의하며 字, 號, 諡號 등은 리터럴로 처리하고 Datatype Property로 연결해 준다. 대표명칭과 字, 號 등의 이름은 동일함을 표시하기 위해 함수 공리를 사용한다. 성명과 관련해서는 초명, 개명, 본명의 관계를 정의할 필요가 있다. 後梁의 太祖 朱全忠²⁸⁵⁾은 원래 이름은 '朱溫'이었으며 唐나라에 귀순하면서 朱全

285) 朱全忠(852년~912년)은 원래 이름은 朱溫이며 黃巢의 부장이었으나 唐나라에 항복하여 '朱全忠'이란 이름을 하사 받았다. 후에 唐나라를 멸하고 後梁을 세우고 '朱浬'으로 개명한다.

忠이란 이름을 하사받고 후에 황제가 되면서 ‘朱晃’으로 改名하게 된다. 張全義²⁸⁶⁾의 경우 본명은 ‘張居言’이고, ‘張全義, 張宗奭’으로 改名했다가 최종적으로 ‘張全義’가 된다. 養子가 되면서 姓氏까지 바꾸는 경우도 있다. 後梁의 朱全忠의 養子인 朱友文²⁸⁷⁾은 원래 姓名은 ‘康勤’인데 朱全忠의 양자로 들어가면서 ‘朱友文’이란 이름을 사용한다. 여기서 初名, 改名, 本名の 관계를 정리하면, 改名은 처음 이름을 기준으로 표현한 것이며, 최종 이름으로 볼 때 이전의 이름은 初名이나 本名이 된다. 또한 改名을 여러번 한 경우에는 중간의 이름은 改名이 되나 마지막 이름은 최종 이름이 된다. 문제는 최종 이름이 익숙하지 않은 경우가 있다는 것이다. ‘朱全忠’이란 이름은 ‘朱濶’에 비해 훨씬 더 널리 사용되고 있다는 점에서 이 경우에 해당한다. 이러한 대중에게 익숙하다는 것은 자의적인 판단기준이 될 수 있기 때문에 본 논문에서는 가급적 최종 이름을 기준으로 관계를 기술하는 것을 원칙으로 하였다.

앞서 한 인물을 지칭하는 명칭은 姓名 이외에도 字, 號, 諡號, 廟號, 堂號, 法號, 法名, 初名, 初字, 本名, 別稱 등이 있음을 언급하였다. 이들이 동일한 인물을 지칭하고 있음을 표시하기 위해서 OWL의 기본 속성 중 속성의 출현 횟수에 대한 제약 속성인 ‘FunctionalProperty’를 사용한다.

OWL의 속성에는 개체들 간의 관계를 기술하는 Object Property와 개체를 데이터 값과 연결하는 Datatype Property가 존재한다. 그런데 앞에서 말한 것처럼 개체와 개체 혹은 개체와 리터럴이 아닌 프로퍼티와 프로퍼티 간에 관계를 표현할 수 있다. 이는 계층관계(rdfs:subPropertyOf), 동치관계(owl:equivalentProperty), 반대관계(owl:inverseOf) 등을 사용하여 다른 속성과의 관계에 의해 공리를 표현할 수 있고, 또한 속성의 출현 횟수를 제한할 수도 있다. ‘함수형 속성(FunctionalProperty)’은 속성의 값의 출현 횟수가 0 혹은 1개로 규정되어 있다. 이때 속성이 함수형 속성이라는 것은 ‘rdf:type’으로 표현한다.²⁸⁸⁾

아래 그림에서 ‘FunctionalProperty’를 지정하면 공리에 의해 출현 횟수가 1회 이하로 제한되므로 같은 주어에 2번 적용하면 목적어가 같다는 것을 의미한다. 즉 ‘項羽’는 ‘項籍, 西楚霸王’으로 불림을 알 수 있다. 이러한 ‘FunctionalProperty’는 클래스와 리터럴 간에도 적용이 가능하다.

286) 張全義(852년~926년)는 원래 이름은 張居言. 字는 國維. 최종 작위는 齊王으로, 諡號는 忠肅王이다. 黃巢의 수하에서 禮部尙書를 지내다가 黃巢가 실패한 후에 唐나라가 투항하여 ‘張全義’란 이름을 하사받았다. 그 뒤 唐나라가 망한 후에 朱全忠의 後梁에 투항하여 ‘張宗奭’이란 이름을 하사 받는다. 後梁이 망한 후에는 다시 李存勳의 後唐에 투항하여 원래 이름인 ‘張全義’로 회복시켜 주기를 청했다.

287) 朱友文은 朱全忠의 養子이다. 朱友文의 아내는 王氏로 미모가 뛰어났는데 朱全忠의 총애를 받았다. 그래서 朱友文은 부인 王氏를 입궁시켜 朱全忠의 시중을 들게 하였다.

288) 神崎正英(저)/황성역·양혜술(역), 앞의 책(2008), 120쪽.

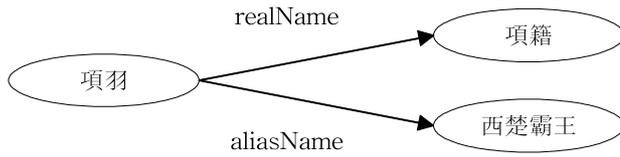


그림 38. 동명이인의 처리

인명 정보를 정리하면 다음과 같다.

표 58. 인명 정보(personInformation)

속성		내용
personInformation		인명정보
	personName	인물의 이름
	hasAliasName	이칭
	hasChangedName	개명
	hasChildhoodName	아명
	hasCourtesyName	자
	hasDharmaName	법호
	hasPersonName	인물이름
	hasEnfoeffedName	봉작호
	hasFormerName	初名
	hasMeritoriousOfficials	공신호
	hasName	구분하기 어려운 이름
	hasNickName	호, 아호
	hasOfficialName	공식명칭, 대표명칭
	hasPenName	필명
	hasPosthumousName	諡號
	hasRealname	본명
	hasReligionName	종교명
	hasSpellingVariations	철자 이명
	hasTempleName	묘호
	gender	성별
	personId	인물ID
	stringId	상세식별자
	UCI	UCI
	legitimacy	적서 구분
	relationshipOrder	관계의 순서
	recordOrder	등재 순서
	siblingOrder	형제 순서
	spouseOrder	배우자 순서

4) 공간 정보(placeInformation)

공간 정보는 점(point)과 선(line), 그리고 면(polygon)으로 구분할 수 있다. 좌표 상의 한 지점은 경도, 위도로 표현할 수 있으며, 이들 점들의 집합이 선(polyline)이 되며, 선이 모여서 연결이 되면 면(polygon)을 이룬다. 공간 정보는 지명과 관련이 있지만 지명 자체를 의미하는 것은 아니다. 지명은 지도상에서는 하나의 점으로 표현할 수 있으나, 실제로는 일정한 영역을 지니고 있다. 그러므로 지명을 표현하기 위해서는 선이나, 면을 이용해서 표현해야 한다. 공간 정보가 중요한 이유는 역사를 이해하기 위해서는 점점 더 전자지도의 필요성이 증가하기 때문이다. 지도는 상대적인 위치 정보와 거리 정보 등을 시각적으로 표현함으로써 누구나 쉽고 빠르게 이해할 수 있는 효과적인 표현 매체이다.²⁸⁹⁾ 역사서에 등장하는 특정한 지역을 단지 개념적으로만 이해하고 역사서를 읽는 것과, 실제로 지도상에서 구체적인 위치를 파악하고 역사서를 읽는 것은 그 이해의 깊이가 다를 수밖에 없다.

공간 좌표를 표현하기 위해서는 기본적으로 경도와 위도로 표현해야 한다. 이와 관련하여 W3C의 wgs84_pos²⁹⁰⁾에 이미 정의되어 있으므로 이 용어를 사용한다.²⁹¹⁾ 여기에는 지리 정보 표현을 위해 필요한 여러 가지 프로퍼티가 있으나 그 중에서 'geo:lat, geo:long'를 사용하고, 여기에 gisURL, polygon, polyline을 추가하였다. gisURL은 'China Historical GIS'²⁹²⁾ 등과 같이 공간 관련 URI 주소를 외부에 제공하는 사이트가 있는데, 이는 이들 URI 주소와 연결시키는 역할을 한다.

공간 좌표 geo:lat, geo:long'는 정의역이 Place가 되고 치역이 숫자 표현 형식인 float이 된다. 사람이 태어나고 죽은 구체적인 지명은 사람과 Object Property로 연결하는 데 비해, 경·위도 좌표는 지명과 Datatype Property으로 float 형식의 숫자와 연결한다.

289) 황인혁, 「조선시대 『濬源寶鑑』의 분석과 활용에 관한 연구」, 건국대학교 대학원 박사학위논문, 2015, 159쪽.

290) http://www.w3.org/2003/01/geo/wgs84_pos

291) W3C Semantic Web Interest Group, W3C, <https://www.w3.org/2003/01/geo>

292) China Historical GIS, <http://www.fas.harvard.edu/~chgis>

표 59. 공간 위치 정보

속성		내용
placeInformation		위치정보
	geo:lat	위도
	geo:long	경도
	polygon	면으로 구성된 공간 정보
	polyline	선으로 되어 있는 것
	gisURL	

5) 관직(position)

관직과 관련하여 관직의 문무 구분, 품계의 관계를 설정하도록 하였다. 하위 프로퍼티로 ‘positionName, positionType, rank’를 두었다.

Person과 관직을 연결하기 위해서는 ‘hasCareer’ 오브젝트 프로퍼티를 이용하여 표현하나 관직의 구체적인 속성을 표현하기 위해서는 ‘position’이라는 Datatype Property를 사용한다.

표 60. 관직

속성		내용
position		관직명
	positionType	관직유형(문반/무반)
	rank	품계

라. 어노테이션 프로퍼티(Annotation Property)

OWL에서 주석 정보가 포함된 Annotation Property에 다음과 같이 5가지가 정의되어 있다.

- owl:versionInfo는 버전에 대한 정보를 제공하며, 치역으로 rdfs:Literal을 갖는다.
- rdfs:label은 Class, Property, Individual 등의 이름에 인간이 이해할 수 있는 이름을 추가하기 위해 사용한다. 치역으로 rdfs:Literal을 갖는다.
- rdfs:comment는 설계한 모델에 대해 이해하기 쉽도록 주석을 추가할 수 있다. 치역으로 rdfs:Literal을 갖는다.
- rdfs:seeAlso는 치역으로 URI 참조를 갖는다.
- rdfs:isDefinedBy는 Class, Property, Individual 같은 Ontology Element를 지정하는 Ontology 참조에 사용되는 URI Reference를 치역으로 갖는다.

OWL에서는 Class, Property, Individual 그리고 Ontology 그 자체에 다양한 정보나 Meta-data의 주석을 다는 것을 허용한다. 즉, 주석, 생성 날짜, 저자 또는 Web Page 같은 Resource에 대한 참조를 나타낸다. OWL-Full은 Annotation Property에 대해 어떠한 제약 조건도 두지 않았으나, OWL-DL은 몇 개의 제한을 두고 있다. 예를 들어 Annotation Property의 대한 대상은 반드시 URI 참조의 문자 정보이거나 Individual이어야 하며, Annotation Property는 Property Axiom에서는 사용할 수 없다. 그리고 그것들은 하위 프로퍼티를 가질 수도 없고, 다른 Property의 하위 프로퍼티가 될 수도 없으며, 정의역과 치역도 설정할 수 없다.²⁹³⁾

본 논문에서는 RDFS 표준에서 제공하고 있는 어노테이션 프로퍼티 중에서 ‘rdfs:seeAlso, rdfs:isDefinedBy’를 이용하여 위키피디아와 같은 관련된 지식을 참조할 수 있도록 하였다. ‘rdfs:seeAlso’는 보통 엔티티를 설명하는 데 도움을 줄 수 있는 다른 문서들이 있다는 것을 의미한다. 그 대상이 위키피디아 항목의 URL 주소가 될 수도 있고, 다른 RDF나 링크된 정보를 포함하는 다른 RDFS가 될 수도 있다. 이런 경우를 위해 ‘rdfs:seeAlso’는 부가적인 정보의 웹 위치를 표시하는 방법을 제공한다. 즉 그것은 인간 가독형이 아닌 URI이어야 한다. ‘rdfs:seeAlso’는 형식적인 시맨틱을 가지지 않으므로 어떤 프로세서가 ‘rdfs:seeAlso’를 만났을 때 정확한 동작이 기술되어 있지 않다. 링크를 만난 툴의 일반적인 동작은 브라우저나 애플리케이션 인터페이스에서 그 링크들이 사용되고 있는 RDFS 문서를 통해서 보여주는 것이다.²⁹⁴⁾ ‘rdfs:seeAlso’와 같이 ‘rdfs:isDefinedBy’는 자원에 대한 정보의 원천에 대한 링크를 제공한다. rdfs:isDefinedBy는 주제 자원을 정의하는 자원을 나타내는 데 사용되는 rdf:Property의 인스턴스이다. 이 속성은 자원이 기술된 RDF 어휘를 나타내기 위해 사용될 수 있다. 주로 모델러들에게 자원의 정의에 대한 기술을 어디에서 찾을 수 있는지 서술하도록 한다.

rdfs:label은 사람이 이해할 수 있는 레이블을 추가할 수 있는 기능을 제공한다. 시맨틱 자원은 URI로 기술되는데 이것은 자원에 대한 범용적인 식별자를 제공하나, URI는 사람이 이해하기 어렵다. RDFS에서 기본적으로 제공하는 프로퍼티 rdfs:label을 사용하여 어떤 자원에 대한 인쇄 가능한 이름을 제공하는 데 사용할 수 있도록 하였다. rdfs:label은 특히 URI 참조에 공백을 포함할 수 없으므로 URI 참조에서는 공백이 없이 표현하되, rdfs:label을 이용하여 공백을 포함한 정확한 이름을 표현하는 데 활용되기도 한다. 특히 rdfs:label에는 언어 태그를 추가하여 기술할 수 있다.

293) OWL Web Ontology Language Reference, W3C 홈페이지, <https://www.w3.org/TR/owl-ref>

294) Dean Allemang·James A. Hendler(저)/김성혁 외(역), 앞의 책(2008), 129쪽.

5. 기존 온톨로지 재사용 검토

특정 분야의 온톨로지를 개발할 때 누군가가 먼저 만들어 놓은 것이 있다면 그 결과가 좋고 나쁨을 떠나서 참조할 만한 가치가 있다. 그것은 최소한 선구자들의 고민선상에서 출발할 수 있기 때문일 것이다. 기존 온톨로지의 재사용은 어휘(vocabulary)의 재사용과 자원의 재사용으로 구분할 수 있다. 온톨로지에 사용하는 어휘는 기존에 발행된 어휘를 재사용함으로써 가치를 가진다. 왜냐하면, 발행된 어휘는 많은 사람이 사용할수록 그 가치가 커지며, 그 결과 더 많은 사람이 그 어휘를 사용하게 되는 선순환 효과를 가져 오기 때문이다. 이것은 곧 새로운 URI를 만들어내는 것이 좋은 것이 아니라, 누군가 다른 사람이 만들어 놓은 URI를 재사용하는 것이 좋다는 것을 의미한다.²⁹⁵⁾ 예를 들자면, 사람들 사이에 친분이 있음을 표시하기 위한 어휘를 새로 만들어 사용할 수도 있겠지만, 기존에 개발된 FOAF의 'http://xmlns.com/foaf/0.1/knows'를 사용하여 관계를 표현하는 것이 온톨로지 개발에 투입되는 시간과 노력을 줄이고 좀 더 완성도가 높은 온톨로지를 개발하는 데 도움을 줄 수 있다.

본 연구에서는 클래스 설계와 프로퍼티 설계 뒤에 기존 온톨로지의 재사용을 기술하였으나, 실제로는 이 세 가지는 동시 혹은 반복적으로 이루어져서 어느 한 단계가 끝나야만 다음 단계로 넘어가는 것은 아니다.



그림 39. 온톨로지 설계 절차²⁹⁶⁾

기존에 개발된 어휘를 참조하기 위해서는 스우글(Swoogle)²⁹⁷⁾나 schema.org와 같은 사이트를 이용하여 기존에 어떤 어휘를 사용하고 있는지 참조하는 것이 좋다. 이들 온톨로지 라

295) 한국정보화진흥원, 「데이터베이스 활용기술전망 부록 :데이터 활용 신기술 가이드 1: RDF」, 2014, 26쪽.

296) 이경일, 「온톨로지 구축과 의미 메타데이터 관리」, 솔트룩스, 2006, 41쪽.

297) Swoogle, <http://swoogle.umbc.edu>

이브리리를 통해 적합한 온톨로지를 검색, 수정해 활용할 수 있다.

Swoogle는 미국 메릴랜드 대학 컴퓨터과학 및 전자공학부 산하의 Ebiquity research group에서 2004년 2월부터 2006년 12월까지 수행하고 있는 연구의 수행 결과물이다. Swoogle은 시맨틱 웹 온톨로지, 시맨틱 웹 인스턴스 데이터를 검색하고, 시맨틱 웹 문서의 메타데이터를 공급하는 역할을 한다.²⁹⁸⁾

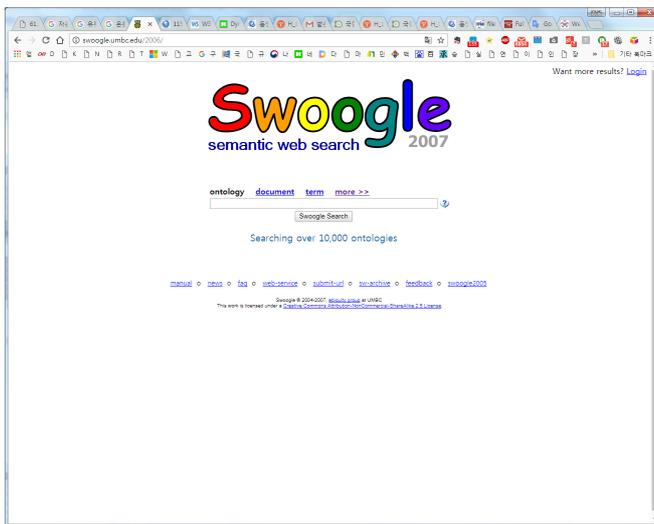


그림 40. Swoogle 시맨틱 검색 엔진

Schema.org는 2011년 6월 2일, Microsoft, Google, Yahoo! 등 대형 검색 엔진 3사가 개설한 웹 사이트로, 2011년 11월에는 러시아에서 가장 큰 검색 엔진 회사인 Yandex가 여기에 참여한다. Schema.org는 웹 페이지 구조적 데이터의 표기를 위한 스키마의 공통 세트를 만들고 지원하는 것이 목적이다. Schema.org에서는 웹 페이지들에 대한 개념이나 속성들을 범용적으로 사용할 수 있도록 스키마들을 공개적으로 제공한다.²⁹⁹⁾ Schema.org는 마이크로 데이터(microdata)라는 포맷을 사용하고 스키마로 인물, 서적, 장소, 이벤트 등 100여 종의 형태(type)와 그 안에서 사용되는 속성(property)을 웹 사이트에 공개하고 있다.³⁰⁰⁾

사람들에게 널리 알려진 Schema.org의 어휘들을 사용해서 기계가 처리할 수 있는 정보들을 웹페이지에 추가하면 검색 엔진들이 보다 명확하게 정보를 처리할 수 있을 뿐만 아니라 또다른 응용 프로그램에 의해서 공개된 데이터의 자동 처리가 가능해진다.³⁰¹⁾

298) 한국교육학술정보원, 「KEM 고도화를 위한 온톨로지 기반 시맨틱 웹 연구」, 2006, 96쪽.

299) 강대현, 이정민, 김태운, 박한샘, 권경락, 정인정, 「메타데이터를 사용한 역사 온톨로지 생성 방법에 관한 연구」, 2015, 801쪽.

300) 강대현 외, 앞의 논문(2015), 801쪽.

온톨로지 어휘를 재활용하기 위해서는 온톨로지가 나타내는 의미와 범주가 정확하게 일치해야 한다. 온톨로지에서는 흔히 外延과 內包를 말한다. 外延은 개념이 가리키는 대상을 말하고 內包는 개념에 포함되는 대상들로부터 추출하여 얻어진 공통속성의 합, 즉 어떤 개념의 내용이 되는 여러 속성을 일컫는 말이다. 간단히 말하면 外延은 집합, 內包는 특성이 다. 본 논문에서도 공간 정보를 표현하기 위한 geo:lat와 geo:long와 서명을 표시하기 위한 프로퍼티로 더블린 코아의 온톨로지를 도입하였다.

「위드넷의 의미 관계 집합을 이용한 온톨로지 매핑」에 따르면 온톨로지의 재활용에는 아래와 같은 문제를 검토해야 한다고 하였다. 첫째, 동의어, 다의어 문제로, 같은 도메인에서 사용되는 동일한 의미의 단어가 서로 다르게 표현되거나, 한 단어가 여러 가지 의미로 사용됨으로써 온톨로지 매핑에 어려움을 주고 있다. 둘째, 동일한 의미로 개념을 정의했지만 동일한 개념으로 인식할 수 없는 경우가 있다. 셋째, 온톨로지 개념에서 하나의 의미를 전달하기 위해 여러 개의 개념을 사용할 수 있다. 예를 들면, 이름 개념을 표현할 때 ‘Name, FirstName, LastName’ 등으로 사용하는 예 등이다. 마지막으로 의미 표현이 개념이나 프로퍼티 모두 가능하다는 것이다. 예를 들면, 한 온톨로지에서는 조언자를 “adviser”라는 클래스로 개념으로 정의하고, 다른 온톨로지에서는 “advisedBy”라는 프로퍼티를 이용하여 정의할 수 있다.³⁰²⁾ 이러한 현실적인 상황을 고려해서 온톨로지를 재활용해야 한다.

일반적으로 온톨로지의 재활용의 순서는 기존의 온톨로지를 분석하여 그것을 적용시킬 수 있을 때 적용하나, 본 논문에서는 반대로 온톨로지 전체를 도출하고 그 결과를 바탕으로 외연과 내포가 일치하는지를 확인하였다.

간혹 어휘를 발행한 후 같은 어휘가 다른 사람에 의해 이미 만들어 있음을 발견할 수도 있다. 이 두 어휘가 같은 것을 의미한다는 것을 표시하는 RDF 트리플을 발행하는 것이며 이때 사용하는 서술어는 owl:sameAs이다.

301) 한국정보화진흥원, 앞의 자료(2014), 15쪽.

302) 광정애·용환승, 「위드넷의 의미 관계 집합을 이용한 온톨로지 매핑」, 『정보과학회논문지 : 데이터베이스』, 제36권, 제6호, 2009, 465~467쪽.

IV. 지식 관계망의 설계

이 장에서는 지식 관계망의 개요에 대해서 설명하고 지식 관계망의 주요한 구성을 이루는 어휘 간 관계를 어떻게 표현할지에 대해서 살펴보았다. 그리고 실제 LOD를 발행하는 단계를 설명한다.

1. 지식 관계망

가. 지식 관계망의 개요

지금까지 설명한 온톨로지 모델을 적용하여 역사적 지식을 LOD로 발행하면 상당수의 지식을 서로 연결할 수는 있겠지만, 학자들의 연구에 의해 정제된 심도 있는 지식의 수준까지 도달할 수는 없다. 그렇기 때문에 辭典 등의 잘 정리된 지식을 LOD로 발행할 필요가 있다. 지식 관계망(Knowledge Network)은 역사 자료의 연계 및 통합을 위해 설계한 온톨로지를 확장 및 보완한 개념으로, 어휘 간 관계에 있어 정제화와 구조화, 계층화라는 세 가지 요소가 추가된 개념이다. 지식 관계망은 하나의 지식 노드와 관련된 다른 노드를 서로 연결해서 분절화된 지식을 의미적으로 연결해 주는 네트워크를 의미한다.

지식 관계망은 역사적 지식을 통합할 수 있는 모델을 제공하며, 어휘 간의 관계를 의미적으로, 내용적으로 엮을 수 있게 한다. 어휘 간의 관계를 의미적으로는 同義語, 類義語, 反意語 등 의미에 따른 관계를 엮어 주는 것이고, 내용적으로는 項羽의 字는 羽고 名은 籍이고, 愛妾은 虞姬이며, 누구와 천하를 놓고 다투었는지 등의 지식 간의 관계를 엮어 주는 것이다. 이러한 관계는 의미에 따라서 관계가 형성된 것이 아닌 내용 속에서 관계가 만들어진 것이다. 즉 項羽와 項羽와 관련된 정보는 필연적이지 않고 역사서의 내용에 의해 만들어진 관계인 것이다. 이러한 관계는 지금까지 연구된 학술적 성과 혹은 정제된 지식을 바탕으로 구축된다. 즉, 지식의 寶庫인 辭典 등에서 수집한 어휘 간의 관계, 同義語, 反意語 등의 의미적 연관 관계와 역사의 주체인 인물 간의 관계, 인물과 사건 간의 관계 등을 정밀하게 정제하여 RDF 트리플로 구축하는 것이다. 간단히 말해 이는 역사서에서 얻을 수 없는 정제된 지식을 사전을 이용하여 보완하고 구조화(Structureing)한다는 의미이다. 계층화는 어휘 간의 텍사노미를 잘 형성하여 논리적 추론을 강화하고 대체 표현을 찾는 데 도움이 될 수 있다. 지식 관계망은 기존 지식을 관련 있는 지식끼리 잘 엮어서 정리하고, 분류하고, 구조화시킨 지식의 보고라고 할 수 있다. 이는 내부의 지식과 외부의 지식을 소통시키는 역할을 한다. 지식 관계망을 통해서 관련된 외부 지식으로 이동하고, 반대로 외부 노드에서 지식 관계망

을 매개로 하여 관련된 지식으로 접근할 수도 있다. 이러한 지식 관계망을 통해서 기존 LOD로 해결할 수 없는 훨씬 더 다양하고 복잡한 지식으로 접근이 가능하게 될 것이며 나아가 새로운 정보로 접근할 수 있는 소통의 교점이 된다.

지식은 점차 한 곳에 집중되고, 통합되는 경향이 있다. 그것은 정확한 정보를 쉽게 찾고 싶다는 필요성에 부응한 결과이다. 지식 관계망은 기구축된 어휘 정보를 의미 중심으로 연계된 통합 서비스를 제공한다. 지식 관계망에서는 같은 개념이 서로 다르게 사용되는 용어, 용어 간의 의미가 일치하지 않음으로써 발생하는 정보의 단절을 방지하고, 특정 개념에 대해 더욱 다양하고 폭넓은 지식으로 접근할 수 있게 해 준다.

나. 시간 관계(hasTime-Span)

역사 연구에서 연도를 정확하게 파악하는 것은 사건을 이해하는 데 중요한 역할을 한다. 만약 정확한 연도를 파악하지 못한다면 역사의 해석이 달라질 수 있기 때문이다.

중국에서는 시간의 기록을 紀年法, 紀月法, 紀日法으로 표현하였다. 고대인들은 간지로 紀日을 했는데 甲·乙·丙·丁 등의 10개의 天干과 子·丑·寅·卯 등의 12개의 地支를 차례대로 조합하여 60단위가 되는데 이를 六十甲子라고 한다. 이러한 紀日法은 甲骨文 시대에도 이미 있었다고 한다.³⁰³⁾ 干支紀年法은 일반적으로 後漢 때부터 시작된 것으로 보는데, 60갑자가 일주하였다가 다시 시작하는 것이 현재까지도 중단되지 않고 계속되고 있다. 일반적인 역사 연표에 기록되어 있는 西漢 이전의 간지는 후인들이 逆推算하여 정리한 것이다. 이런 六十甲子 이외에도 皇帝의 在位年과 年號로 연도를 표현하기도 하였다. 年號는 군주가 자신의 治世年次에 붙이는 칭호로 多年號 또는 元號라고도 하였다. 前漢의 武帝가 年號를 사용하기 전까지는 왕의 在位年이 연도를 세는 기준으로 사용하다가 漢 武帝가 建元이라는 연호를 사용하면서 비로소 年號를 이용하여 연도를 표현하기 시작했다.

이러한 年號는 중국뿐만 아니라 한국, 일본, 베트남 등의 국가에서도 사용되었다. 그렇기 때문에 중국 연호와 우리나라 연호, 일본 연호를 상호 연계시키기 위한 온톨로지 설계가 필요하다. 무엇보다도 年號는 王朝와 皇帝(왕), 간지 등과 서로 연결이 되어야 한다.

먼저 시간을 나타내는 年號, 왕대력 등의 특징에 대해서 간단하게 살펴보기로 한다.

첫째, 年號는 일정한 시간 구역을 가지고 있다. 즉 시작 연도와 끝나는 연도가 있다. 예를 들어 太興이라는 年號는 서기 318년부터 321년까지 東晉의 元帝(司馬睿)가 다스렸던 기간을 말한다. 둘째, 年號는 서로 다른 왕조에서 같은 연호를 사용하기도 하였으며, 한 임금이 여러 개의 연호를 사용하기도 하였다. 東晉의 惠帝 司馬衷은 재위 기간 중 永熙(290), 永平

303) 王力(지)/李鴻鎮(역), 『中國古代文化常識』, 형설출판사, 1989, 24쪽.

(291), 元康(291~200), 永康(300~301), 永寧(301~302), 太安(302~303), 永安(304), 建武(304), 永興(304), 光熙(306) 등의 연호를 사용하였다. 그러다보니 연호로 인한 많은 혼란이 발생하게 되었다. 明의 太祖 朱元璋은 一世一元制를 수립하게 되면서 이런 혼란을 줄여 들었다. 이 원칙은 正統帝 朱祁鎮의 연호가 正統과 天順으로 두 개인 것을 제외하고는 청나라 말기까지 잘 지켜졌다.³⁰⁴⁾ 셋째, 같은 시기에 여러 왕조가 있음으로 인해 여러 개의 연호가 같은 시기를 나타내기도 한다.

아래 표를 보면 서기년도에 대응하는 왕조별 연호가 있다. 어떤 연호가 나오더라도 서기년을 기준으로 동일한 시간을 추론할 수 있어야 한다.

표 61. 동 시대 복수의 왕조의 예

서기	晉		成漢		漢趙		後趙	
317	元帝	建武1	太宗武皇帝	玉衡7	昭武帝	麟嘉2		
318	元帝	太興1	太宗武皇帝	玉衡8	昭武帝/隱皇帝/靳準/靳準	麟嘉3/漢昌1/-/光初1		
319	元帝	太興2	太宗武皇帝	玉衡9	劉曜	光初2	高祖明皇帝	太和1
320	元帝	太興3	太宗武皇帝	玉衡10	劉曜	光初3	高祖明皇帝	太和2
321	元帝	太興4	太宗武皇帝	玉衡11	劉曜	光初4	高祖明皇帝	太和3
322	元帝	永昌1	太宗武皇帝	玉衡12	劉曜	光初5	高祖明皇帝	太和4
323	元帝/明帝	永昌2/太寧1	太宗武皇帝	玉衡13	劉曜	光初6	高祖明皇帝	太和5

年號의 연결과 관련하여 생기는 문제는 서로 다른 왕조가 동일한 年號를 사용하는 경우이다. 예를 들어 중국 역사상 建元이라는 年號는 西漢, 前趙, 東晉, 前秦, 南齊에서 사용했으며, 우리나라의 新羅에서도 사용했었다.

304) 明 英宗은 연호가 正統과 天順 등 2개가 있다. 영종은 오이라트 족을 정벌하러 친정을 했다가 토목에서 적에게 포로가 된다. 그러자 명나라에서는 正統帝의 이복동생인 景泰帝를 황제로 옹립한다. 오이라트는 포로로써 가치가 없어진 英宗을 조건없이 돌려보낸다. 英宗은 나중에 正統帝를 중심으로 하는 일파가 정변을 일으키면서 天順帝라는 이름으로 복위하게 된다.

표 62. 동일한 年號를 사용하는 王朝의 예

年號	王朝	식별자	廟號	期間
建元	西漢	建元_西漢	武帝(劉徹)	B.C. 140–B.C. 135
建元	前趙	建元_前趙	昭武帝劉聰	A.D. 315–A.D. 316
建元	東晉	建元_東晉	康帝(司馬岳)	A.D. 342–A.D. 344
建元	前秦	建元_前秦	宣昭帝(苻堅)	A.D. 365–A.D. 385
建元	南齊	建元_南齊	高帝(蕭道成)	A.D. 479–A.D. 482
建元	新羅	建元_新羅	法興王, 眞興王	A.D. 536–A.D. 551

그렇다면 각각의 연호를 구분해 주기 위해서 어떤 방법을 써야 할까? 우선 URI 참조를 어떤 형태로 할지부터 결정을 해야 한다. 온톨로지 설계와 달리 LOD 서비스를 위해서는 URI 주소로 실제로 접속이 가능해야 하기 때문에, 온톨로지 설계 단계에서는 고민하지 않았던 문제가 새롭게 발생한다. URI 참조에 관해서는 앞서 다루었으므로 여기서는 구체적인 방법에 대해서만 다루기로 한다. 연호를 식별하기 위해서는 먼저 단순 일련번호를 이용하여 표현하는 방법과 建元_西漢처럼 연호에 구분자를 추가하여 구분하는 방법, 그리고 URI 주소를 이용하여 구분해 주는 방법이 있을 수 있다.

첫 번째 방법은 전체 개체에 중복을 제거한 후 일련번호를 부여하여 식별하는 방법이다. 그러나 이 방법은 기계에 유리한 방법으로 사람이 이해하기에는 어렵다. 그렇기 때문에 사람들이 이해할 수 있는 레이블(rdfs:label)을 첨부하여 사람이 이해할 수 있도록 한다. 또한 관계형 데이터베이스에 기본키(PrimaryKey)가 있을 경우에 손쉽게 변환할 수 있다는 장점이 있다. 두 번째 방법은 연호와 구분자를 결합한 형태로 URI 참조를 만드는 방법이다. 여기서 기본 URI를 'http://dh.aks.ac.kr/ontologies/historicevent/kno'로 한다면 # 뒤에 연호와 구분자를 추가하면 된다. URI 주소에는 공백을 사용할 수 없기 때문에 공백 대신에 괄호, 언더 바 등을 사용할 수 있다. 세 번째 방법은 URI 주소에 다른 연호와 구분해 줄 수 있는 경로를 넣고 이를 네임스페이스를 사용하여 표현하는 방법이다. 다음의 표에서는 연호가 중첩되지 않게 중간에 왕조명인 '後蜀'을 로마자 'Huchok'로 변환하여 넣어 주었다. 이처럼 URI 주소는 전체 경로를 사용해도 되지만, 그럴 경우에 복잡해지는 문제가 있기 때문에 이를 간편하게 표현하기 위해서 보통 네임스페이스를 사용한다. 그런데 네임스페이스를 사용할 경우에 왕조의 수 혹은 구분해 주어야 할 왕조의 수만큼의 네임스페이스를 만들어야 한다는 문제가 있다.

이 세 가지 방법 중에 어느 하나를 선택할 때 또 하나 고려해야 할 것은 SPARQL에서 조회와 인터링킹의 편의성이다. LOD라는 것이 기계가 이해할 수 있는 웹을 지향하는 것은 사실이나 모든 것을 다 컴퓨터가 알아서 해결해 줄 수는 없는 일이다. 즉 사람이 분석하고

그 결과를 학습시키는 과정이 필요한 데 만약 네임스페이스를 왕조별로 구분한다면 정확한 내용을 파악하는데 쉽지 않을 것이다. 이런 현실적인 문제를 감안하여 본 연구에는 두 번째 방법을 채택한다.

표 63. 중복되는 개체명의 식별 방안

구분	방법	예시
단순 일련번호와 label을 이용	단순한 일련번호로 구분하고 rdfs:label로 참조할 수 있게 하는 방법	<pre><owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/kno#EraName_0001"> <rdf:type rdf:resource="&kno;EraName"/> <rdfs:label xml:lang="kr">明德</rdfs:label> </owl:NamedIndividual></pre>
연호+구분자	연호와 연호를 구분할 수 있는 왕조나 시대를 결합하는 방법	<pre><owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/kno#建元_西漢"> <rdf:type rdf:resource="&kno;EraName"/> <rdfs:label xml:lang="kr">明德</rdfs:label> </owl:NamedIndividual></pre>
URI 경로로 표현	URI 경로 자체에 구분할 수 있는 경로를 넣어주는 방법	<pre><owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/EraName/Huchok/kno#建元"> <rdf:type rdf:resource="&kno;EraName"/> <rdfs:label xml:lang="kr">明德</rdfs:label> </owl:NamedIndividual></pre>

중복되는 연호를 각 왕조별로 구분하였으면 서기년을 기준으로 하여 시점과 종점을 연결해 주어야 한다. 시간의 연계는 반복되지 않은 기준인 서기연도와 일정한 구간을 가지는 왕조별 연호와 연차, 그리고 60년을 단위로 반복되는 간지를 서로 연계시켜야 한다. 예를 들어 서기년을 기준으로 서기 317년에 해당하는 建武1, 玉衡7, 麟嘉2 등은 서로 동일한 시간대를 표시한다. 이러한 연호에 의한 연도의 표현이 서기년을 기준으로 서로 같은 시간대를 나타내고 있음을 명시적으로 표현해 주어야 한다. 이는 OWL에서 기본적으로 제공하는 프로퍼티인 'owl:sameAs'에 의해서 표현할 수 있다. 이를 RDF 그래프로 표현하면 다음과 같다.

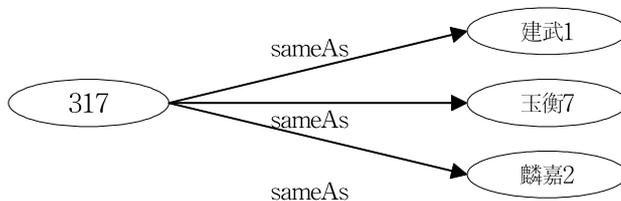


그림 41. 서기년과 연호의 매핑

이를 OWL로 표현하면 다음과 같다.

```
<owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/kno#317">
  <rdf:type rdf:resource="&kno;Year"/>
  <owl:sameAs rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent/kno#建武1"/>
  <owl:sameAs rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent/kno#玉衡7"/>
  <owl:sameAs rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent/kno#麟嘉2"/>
  <kno:sexagenarycycle rdf:resource="&kno;丁丑"/>
</owl:NamedIndividual>
```

‘sameAs’는 OWL에서 기본적으로 제공하는 프로퍼티이며 서로 다른 온톨로지 개체가 실 세계에서 동일한 자원임을 표현해 준다.

표 64. owl:sameAs를 이용하여 서기년(Year)과 연호, 묘호의 연결

관계명	sameAs	
	정의역	치역
범위	Year	EraYear posthumousNameYear templeNameYear
예제	222년	章武2 黃武1 黃初3

이와 같이 일정한 기간을 가지는 것은 年號뿐만이 아니다. 廟號, 王朝, 時代, 사람 등도 일정한 시간의 범위를 가지고 있다.

표 65. 구간을 가지는 시간의 예

구분	예시	
시대	古代, 中世, 近代, 春秋時代, 三代時代	
서력	789년	
왕조	唐(618-907), 秦(B.C.221-B.C.206)	
묘호	唐太宗(599-649)	
묘호+연도	朝鮮太宗2(1402)	
시호	漢武帝(B.C.156-B.C.87)	
시호+연도	高后八年(B.C.180)	
연호	貞元(785-805)	
연호+연도	清泰元年(936), 泰始二年(266), 太平眞君十年(449)	
이름+연호	耶律億六年(912)	
시간	晡時	
계절	春, 夏, 季秋, 季秋	
절기	立春	
月	月	十二月, 閏十月, 正月, 一月, 冬十月
	干支	甲子月
正朔		
日	日	一日, 五日
	干支	甲寅
時間		子時, 晡時, 申時
특정일		往亡日
구간		20년

다음은 왕조를 중심으로 어떤 기간을 가지는지 파악해 본다. 하나의 왕조는 왕조가 시작 되는 시간과 끝나는 시간이 있다. 이러한 정보를 서로 연계시키기 위해서 Object Property 중 'dynastyStart'와 'dynastyEnd'를 사용하여 왕조의 시작하는 연도와 끝나는 연도를 표시 한다. 즉, 朱全忠이 건국한 '後梁'은 시작 시점은 'dynastyStart'를 이용하여 'Year' 클래스의 913과 연결하고, 끝나는 시점은 'dynastyEnd'를 이용하여 923과 연결하는 방식이다.

표 66. 왕조와 관련된 정보의 연결

	구분	예	프로퍼티
後梁	시대	五代十國	hasPeriod
	왕조시작	913	dynastyStart
	왕조끝	923	dynastyEnd
	황제	朱全忠, 朱友貞	emperor
	수도	洛陽, 開封	capital
	주요사건	後梁太祖即位	hasEvent
	관련정보	위키, 네이버, 바이두	owl:seeAlso

이를 OWL로 표현하면 다음과 같다.

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/Dynasty/kno#後梁 -->

<owl:NamedIndividual rdf:about="&knoDynasty;後梁">
  <rdf:type rdf:resource="&kno;Dynasty"/>
  <rdfs:seeAlso rdf:datatype="&xsd:string">http://baike.baidu.com/link?url=1mkJUHUBP7GwBm
AZw0J4oTYght1F_nCv2xJwCqE7Iy1K5nzp_G0D2Y2Tzv5spXvxUeLb91ELSq-Mz_yy33WJpDPob
Fci6C3fYM5GTAUtUAG</rdfs:seeAlso>
  <rdfs:seeAlso rdf:datatype="&rdfs:Literal">https://zh.wikipedia.org/wiki/%E5%90%8E%E6%A
2%81
22222222</rdfs:seeAlso>
  <kno:emperor rdf:resource="&knoPerson;後梁太祖朱全忠"/>
  <kno:emperor rdf:resource="&knoPerson;後梁末帝朱友貞"/>
  <kno:capital rdf:resource="&knoPlace;洛陽"/>
  <kno:capital rdf:resource="&knoPlace;開封"/>
  <kno:dynastyStart rdf:resource="&kno;913"/>
  <kno:dynastyEnd rdf:resource="&kno;923"/>
</owl:NamedIndividual>
```

後梁의 황제는 後梁太祖朱全忠과 後梁末帝朱友貞의 재위년이나 왕의 순서 등은 해당 황제의 정보에서 order를 이용하여 표현한다.

다음으로 사람도 생년과 몰년이 있어 일정한 기간을 가지는 것으로 볼 수 있으나 일반인의 생몰년은 시간을 파악할 때 황제의 재위 기간만큼 많이 사용되지 않는다. 그렇기 때문에 본 설계에서는 포함시키지 않았다.

표 67. dynastyStart

관계명	dynastyStart/dynastyEnd	
	정의역	치역
범위	dynasty	hasPeriod eraNameYear posthumousNameYear dynastyStart templeNameYear

지금까지의 시간 관련 정보를 정리하면 다음과 같다.

표 68. hasTime-Span

속성		내용
hasTime-Span		위치정보
	dynastyStart	왕조가 시작되는 시간
	dynastyEnd	왕조가 끝난 시간
	reignStart	재위년이 시작하는 시간
	reignEnd	재위년이 끝나는 시간
	hasPeriod	시대를 가지다
	hasEraYear	연호를 가지다
	eraStart	연호가 시작되는 시간
	eraEnd	연호가 끝나는 시간
	sexagenarycycle	해당 시간이 나타내는 간지
	sameyear	연도

다. 어휘 관계(termRelation)

1) 개념적 연관 관계

어휘 간 관계는 본문 내용의 표현과는 관계없이 순수하게 의미적인 동등 관계, 계층 관계나 연관 관계를 표현해야 하는 경우에 사용한다. 의미적 동등 관계는 同義語, 類義語를 말하며, 의미적으로 반대되는 反意語가 있다. 계층 관계와 연관 관계는 역사서에서 지식을 표현하는데도 적용할 수 있지만 辭典 등의 정제된 지식을 참조하여 지식을 구조화한 지식 관계망을 구축하는 데 더 중요한 역할을 한다. 이러한 관계는 일종의 시소러스를 확장시킨 개념으로 볼 수도 있고, 어휘 온톨로지의 일종으로도 볼 수 있다. 시소러스에서는 동등관계를 USE와 UF 등으로 표현하는데 본 논문에서는 명시적으로 동의 관계(hasSynonym), 반의 관계(hasAntonym)를 표현해 주는 것으로 하였다. 어휘 간 계층 관계와 연관 관계에 대한 온톨로지 구조는 주로 『구조적 학술용어사전 기반 온톨로지 구축론』³⁰⁵⁾을 참조하였다. 앞의 책에서는 계층 관계를 속-종 관계, 전체-부분 관계로 나누었고, 연관 관계를 개념적 연관 관계, 기능적 연관 관계, 시간적 연관 관계, 공간적 연관 관계로 구분하여 상세하게 설명하고 있다. 본 온톨로지 설계에서는 이 중 계층 관계를 도입하였으며 추가적으로 외부 지식과 연계시키기 위한 장치로써 ‘관련지식’을 추가하였다. ‘관련지식’은 한국향토문화전자대전³⁰⁶⁾ 웹 서비스에서 본문 중에 마크업된 문맥 요소와 『한국민족문화대백과사전』 등의 사이트에

305) 고영만·김비연·민혜령·송민선·이승준, 『구조적 학술용어사전 기반 온톨로지 구축론』,

한국도서관협회, 2016, 173~192쪽.

306) 한국향토문화전자대전, 한국학중앙연구원, <http://www.grandculture.net>

서 관련된 부분을 찾아서 연계시키는 서비스를 ‘지식 연계’라는 이름으로 제공하고 있는데 이와 동일한 개념이다. 본 논문에서는 수집된 지식 자원을 웹에서 제공하는 사전과 연계시키거나 원 출처로 이동할 수 있는 서비스라는 점을 염두에 두었다. 즉 개체(인스턴스)와 관련이 있는 외부 지식을 연결시켜 주는 역할을 하며, 주로 백과사전이나 관련 분야 전문 사전, 원문 텍스트 등으로 이동할 수 있게 한다.

가) 동의어 관계(hasSynonym)

엄격하게 따지면 완전한 동의어라는 것은 존재할 수 없다. 그러나 일반적인 의미로 동의어와 유의어를 구분해 보자. 어휘가 다르나 뜻이 같은 경우에 동의어라고 한다. 『通鑑節要』에 ‘旰食累朝’³⁰⁷⁾라는 어휘가 나오는데 이 어휘는 줄여서 ‘旰食’ 또는 ‘宵旰’이라고도 한다. 이 말은 ‘해가 진 뒤에야 저녁밥을 먹는다’는 뜻으로, 임금이 국사에 바빠 겨를이 없음을 비유하는 말이다. 이러한 동의어는 ‘hasSynonym’를 사용하여 표현한다.

표 69. hasSynonym

관계명	hasSynonym	
범위	정의역	치역
	term	term
예제	旰食累朝	旰食, 宵旰

이를 OWL로 표현하면 다음과 같다. ‘旰食累朝’는 고사성어(Idiom)의 type이며 둘은 동의어 관계가 있음을 표현한다.

```
<owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/kno#旰食累朝">
  <rdf:type rdf:resource="&kno:Idiom"/>
  <kno:hasSynonym rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent/kno#旰食"/>
</owl:NamedIndividual>
```

나) 반의어 관계(hasAntonym)

어휘 간 반의어 관계에 있는 것을 표현한다.

307) 『通鑑節要』 권47 唐紀 武宗[癸亥]에 “會昌三年從此로 得併力於西邊 更無虞於南路 憑陵近甸 旰食累朝”라 하여 “吐蕃들이 서쪽 변경에 힘을 한데 모으고 다시는 남쪽 방면에 대해 근심하지 않았습시다. 그리하여 吐蕃이 近畿 지방을 능멸하여 몇 조정이 이로 인해 편안하지 못해서 제때에 밥을 먹지 못했습니다.”

표 70. hasAntonym

관계명	hasAntonym	
범위	정의역	치역
	term	term
예제	攻撃	防禦

이를 OWL로 표현하면 다음과 같다.

```

<owl:NamedIndividual rdf:about="http://dh.aks.ac.kr/ontologies/historicevent/kno#攻撃">
  <rdf:type rdf:resource="&kno:Term"/>
  <hasAntonym rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent/kno#防禦"/>
</owl:NamedIndividual>

```

다) 대역어 관계(translationWord)

한문을 번역할 때 한문의 명사는 대부분 우리말의 명사로 번역되는 반면, 동사는 우리말의 명사에 ‘하다’가 붙어서 번역되는 경우가 많다. 한문의 어휘는 우리말로 번역하는 단계에서 문맥에 맞게 번역하면서 원래 가지고 있는 의미와 다르게 번역되는 경우도 많다. 그렇기 때문에 원문과 대역어 관계는 적절하게 관계를 맺어 줄 필요가 있다. 이런 관계가 축적되면 자동 번역 등에도 활용할 수 있을 것이다. 이러한 어휘 간의 관계를 표현할 때 ‘translationWord’를 사용한다. 대역어는 동의어의 일종으로 볼 수 있지만 본 연구에서는 언어가 다른 경우에만 대역어로 표현한다. 이때 정의역과 치역은 모두 Thing이다. 대역어는 하나의 행위 동사와 이 어휘에 대한 대역어를 연결하는 것이 가장 큰 목적이거나, 대역어에 적용되는 것은 행위 동사뿐만 아니라 모든 어휘에 해당하므로 그 범위는 Thing이 된다.

표 71. 대역어 관계

관계명	translationWord	
범위	정의역	치역
	Thing	Thing
예제	誅 죽다01	주벌하다, 죽이다01, 처형하다 崩, 薨, 死, 殒

2) 계층 관계

지식 관계망에서 계층 관계와 연관 관계는 중요한 역할을 한다. 예를 들어 아래 그림에서 李滉과 관련한 정보에서 같은 시험에 합격한 사람들이 누가 있었는지, 李滉이 어떠한 관직을 역임했는지 알기 위해서는 이들 정보 간에 계층 관계를 맺어 주어야 한다. 여기서 안다는 것은 단순히 사람뿐만 아니라 기계가 이해한다는 것을 의미한다.



그림 42. 李滉을 중심으로 한 계층 관계의 표현

계층 관계에 대해서는 국제 표준이나 국가 표준 등이 있으며 이를 온톨로지로 적용한 연구도 적지 않다. 이러한 관계에 대해서는 『구조적 학술용어사전 기반 온톨로지 구축론』³⁰⁸⁾에 상세하게 제시되어 있으므로 본 논문에서는 이를 대부분 그대로 수용한다.

가) 상위어·하위어 관계

속-중 관계는 하나의 클래스 혹은 범주와 이를 구성하는 요소와의 관계를 말한다. 특정 개념과 해당 개념의 구성원, 혹은 종류 간의 관계가 이에 해당한다. 속-중 관계의 경우 특정 범주 및 해당 범주를 구성하는 관계에 해당하는 완전 속-중 관계와 사물, 행위, 행위자

308) 고영만 외, 앞의 책(2016), 174-177쪽. 가) 속-중 관계로부터 개념사레관계까지 모두 온톨로지 설계를 그대로 받아들였으며 예시는 본 논문의 성격에 맞도록 수정하였다.

의 유형에 적용되는 유사 속-종 관계로 나누어 볼 수 있다. 속-종 관계는 용어 간의 개념을 상위개념 및 하위개념으로 구분하여 논리적 계층구조에 따라 전개하기 위한 관계 설정을 의미한다.

(1) 계층 일반

계층 관계를 속-종 관계, 전체-부분 관계, 특정 개념과 사례의 관계와 같이 상세화할 수 없는 경우에 이를 포괄하는 기본적인 계층 관계는 상위어(BT)와 하위어(NT)로 상정한 다.³⁰⁹⁾ 여기서의 상위어와 하위어를 SKOS에서 사용하는 broader, narrower를 사용하지 않은 이유는 BT(NT)는 아래에 나오는 여러 가지 유형으로 표현하기 어려운 것을 대상으로 하기 때문에 SKOS에서 정의한 개념보다는 그 범위가 적기 때문이다. 정의역과 치역은 모두 Thing이다.

표 72. BT(NT)

관계명	BT(NT)	
범위	정의역	치역
	Thing	Thing
예제	死	崩

(2) 종류(유형)

정의역은 Thng 중에 어떤 기준에 따라 나눈 갈래(種類, 類型, 種)를 의미하며, 정의역은 치역에 대해 종류의 관계를 가진다.³¹⁰⁾ 예를 들어 唐律에 있는 형벌의 종류로는 笞刑, 杖刑, 徒刑, 流刑, 死刑이 있다. 이때 ‘唐刑의 종류에는 笞刑이 있다’가 성립한다.

309) 고영만 외, 앞의 책(2016), 174쪽.

310) 고영만 외, 앞의 책(2016), 175쪽.

표 73. hasKind(isKindOf)

관계명	hasKind(isKindOf)	
범위	정의역	치역
	Term	Term
예제	무형 문화재 唐刑	중요무형 문화재 笞刑, 杖刑, 徒刑, 流刑, 死刑

나) 전체-부분 관계

유로워드넷에서는 전체-부분 관계를 다음과 같이 다섯 개의 하위 유형으로 나누었다. 첫째, 전체(whole)와 구성 요소(constituent parts)를 나타내는 관계, 둘째, 집합(set)과 그 요소(members)를 나타내는 관계, 셋째, 대상(object)과 그것을 구성하는 재료(the substance it is made of)를 나타내는 관계, 넷째, 전체(whole)와 그것의 일부분(a portion of it)을 나타내는 관계, 다섯째, 장소(place)와 그 안에 포함된 위치(location included within it)를 나타내는 관계로 나누었다.³¹¹⁾ 그런데 유로워드넷에서 위와 같이 전체와 부분 관계를 구별하는 이유는 모든 언어에서 데이터가 도출되는 출처로서 자원이 ‘amount of, member of, group of, place where, made of, which contains’와 같은 구체적인 패턴과 구조를 포함하고 있다고 생각했기 때문이다.³¹²⁾ 이들을 참조하여 전체와 부분 관계를 정리하면 다음과 같다. 전체-부분 관계는 물리적인 실체가 있는 것과 없는 개념적인 것으로 구분할 수 있다. 물리적인 실체가 있는 것으로는 자연(인공)적으로 만들어진 물질과 재료 간의 관계, 지리적인 위치나 공간 범위를 갖는 대상 간의 관계, 身體와 身體를 구성하고 있는 組織이나 器官과의 관계 등이 있으며, 물리적인 실체가 없는 개념적인 관계로는 집단과 구성원의 관계 등을 들 수 있다.

(1) 분과·계통의 관계

분과·계통의 관계는 정의역으로부터 갈라져 나와 계통을 이루는 관계를 강조하고자 할 경우에 사용한다. 상위 사상, 이론, 학문, 법률 등으로부터 갈래를 이루게 되는 하위 용어에 적용한다.³¹³⁾ 이러한 사례는 많지 않을 것 같으나 우선은 포함시켜 어휘 관계에 표현할 수

311) Vossen Piek(편)/한정환 외(역), 앞의 책(2004), 68쪽.

312) Vossen Piek(편)/한정환 외(역), 위의 책(2004), 68쪽.

313) 고영만 외, 앞의 책(2016), 175쪽.

있게 하였다.

표 74. hasBranch(isBranchOf)

관계명	hasBranch(isBranchOf)	
범위	정의역	치역
	Thing	Thing
예제	法家	性惡說

(2) 전체와 구성 요소와의 관계

전체와 구성 요소와의 관계는 구성 요소가 모두 합쳐져서 하나의 전체를 구성하는 관계이다. 이때 구성 요소는 전체에 포함되더라도 자신의 정체성을 그대로 유지하고 있다. 전체를 구성하는 구성 요소는 그 범위가 한정되어 있으며 이들을 모두 합해야 전체를 지칭하는 명칭이 성립한다.³¹⁴⁾ 예를 들어 三代는 중국의 夏·殷·周를 의미하는데 이 三代라는 개념을 구성하고 있는 구성 요소는 夏·殷·周이다. 이때 夏·殷·周라는 개별 구성 요소는 각각을 지칭하여 三代라고 부르지는 않는다. 즉 夏나라가 三代의 일부를 구성하고는 있으나 夏나라 단독으로는 三代라 칭하지는 않으며, 이들 세 나라가 모두 합쳐져야 비로소 삼대가 되는 것이다. 이는 王朝라는 개념과 夏·殷·周라는 구체적인 사례의 관계와 비교하면 夏·殷·周는 각각이 王朝로서 자격을 갖추고 있다. 즉 이들은 모두가 왕조의 구체적인 사례인 것이다. 이러한 전체와 구성 요소와의 관계에는 總稱과 併稱이 포함된다. 十角은 漢代에 南匈奴 左右賢王·左右谷蠡王·左右日逐王·左右溫禺鞮王·左右漸將王 등의 열 가지 爵位에 대한 總稱인데 이들도 전체와 구성 요소의 관계이다. 牛·李는 당나라 말기의 牛僧孺와 李宗閔의 併稱인데 모두 전체와 구성 요소의 관계이다.

314) 고영만 외, 위의 책(2016), 175쪽.

표 75. hasComponent(isComponentOf)

관계명	hasComponent(isComponentOf)	
범위	정의역	치역
	Thing	Thing
예제	五教兩宗 松都三絶 五禮 五代十國	戒律宗·法相宗·涅槃宗·法性宗·華嚴宗·曹溪宗·天台宗 徐敬德·黃眞伊·朴淵瀑布 吉禮·凶禮·軍禮·殯禮·嘉禮 後梁·後唐……

(3) 집단과 구성원의 관계

집단과 구성원의 관계는 하나의 집단과 집단을 구성하고 있는 구성원 간의 관계를 의미하며 'hasMember(isMemberOf)'를 이용하여 표현한다.³¹⁵⁾ 여기서 집단은 官署나 器官, 團體, 組織, 그룹, 黨派, 政黨 등을 의미한다. 개념과 사례와 관계와 비교해보면 사례는 하나하나가 모두 개념을 충족시키는 관계이다. 예를 들어 後梁, 後唐 등의 나라가 있는데 이들 개별 나라들은 모두 王朝라는 개념의 구체적인 사례에 해당한다. 이에 비해 조선 시대 中宗甲午式年試라는 항목을 보면 이 시험에 합격한 합격자는 金希聖, 丁應斗, 金三宅 등이 있는데 이들은 中宗甲午式年試에 합격한 합격자이지만, 이들 하나하나가 中宗甲午式年試의 개념을 충족시키는 않는다.

표 76. hasMember(isMemberOf)

관계명	hasMember(isMemberOf)	
범위	정의역	치역
	Agent	Person
예제	內命婦 議政府 中宗甲午式年試 推忠協贊靖難開國定社功臣	妃, 嬪, 貴人, 尙宮 領議政, 左議政, 右議政 金希聖, 丁應斗, 金三宅… 太宗, 李和, 李芳毅, 李芳幹, 李佇…

(4) 자연물과 자연물을 구성하고 있는 물질의 관계

자연적으로 만들어진 생명체나 화합물 등의 자연물과 이를 구성하는 물질의 관계에 적용

315) 고영만 외, 앞의 책(2016), 176쪽.

한다. 정의역은 Object이며 그 중에 자연물이며 치역은 정의역을 구성하는 구체적인 물질이나 재료이다. 일단 정의역을 구성하면 치역은 자신의 고유한 정체성을 잃어버릴 수도 있다.³¹⁶⁾

표 77. containsSubstance(isSubstanceOf)

관계명	containsSubstance(isSubstanceOf)	
범위	정의역	치역
	Object	Object
예제	체지방 수정란	피하지방 정자

(5) 인공물과 재료와의 관계

인공물을 만드는 데 필요한 물질이나 재료 간의 관계에 적용한다. 치역은 인공적으로 만들어진 정의역의 Object이다. 그 중에서도 구체적으로 원래 성분이나 만들어진 재료이다. 일단 정의역이 되면 치역은 자신의 정체성을 잃어버리거나 원재료와는 다른 형태로 변화한다. 예를 들어 『通鑑節要』에서 孟嘗君이 秦나라 昭王에게서 탈출하기 위해 昭王이 총애하는 姬에게 바친 狐白裘는 여우의 겨드랑이 부분의 흰 털이 있는 가죽으로 만든 겹옷으로 매우 귀한 옷이었다. 狐白裘와 狐皮는 인공물과 재료와의 관계이다.

표 78. hasIngredient(isIngredientOf)³¹⁷⁾

관계명	hasIngredient(isIngredientOf)	
범위	정의역	치역
	Object	Object
예제	狐白裘	狐皮

(6) 공간적 부분 관계

공간적인 범위를 갖는 대상 간의 관계에 적용한다. 공간적인 범위는 행정 구역이나 지명을 의미한다. 치역은 정의역에서 구분할 수는 있지만 별도로 분리할 수는 없는 관계이다. 이런 공간적 부분 관계는 전이성을 가진다.

316) 고영만 외, 앞의 책(2016), 176쪽.

317) 고영만 외, 앞의 책(2016), 176쪽.

표 79. spatiallyInclude(spatiallyInside)

관계명	spatiallyInclude(spatiallyInside)	
범위	정의역	치역
	Place	Place
예제	秦 唐 京畿道	咸陽 京畿道, 關內道, 都畿道, 河南道 京兆府, 華州, 同州, 坊州, 丹州...

唐나라의 지방 행정 제도를 보면 道-州-縣의 3층위로 구성되어 있다. 초기에 10개의 道로 구분하였다가, 玄宗 때 전국을 15道도 나누고, 道 아래에는 府·州·縣으로 나누어, 각각 府尹·刺史·縣令이 관장하도록 하였다. 이들 관계를 그림으로 표현하면 다음 그림과 같다. 王朝(Dynasty)와 唐의 관계는 아래에서 설명하는 개념과 사례의 관계이며, 唐과 당나라를 구성하고 있는 행정 구역인 京畿道와의 관계는 공간적 부분 관계이므로 spatiallyInclude로 표현한다. 역 관계는 spatiallyInside로 표현한다.³¹⁸⁾

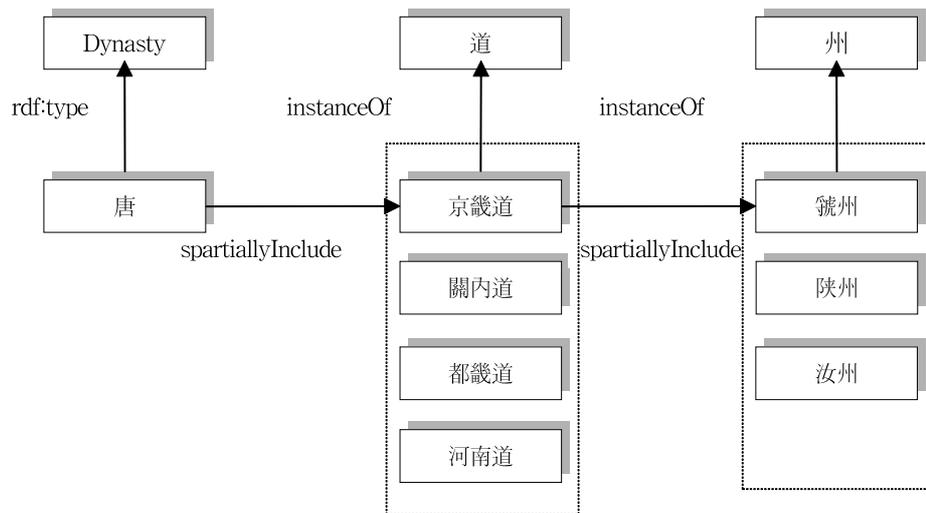


그림 43. 공간적 부분 관계 - 당나라 행정 구역

318) Wang, Yandong, Jingjingf Dai, Jizhen Sheng, Kai Zhou, Jianya Gong, 「Geo-ontology design and its logic reasoning」, 『Geoinformatic 2007: Geospatial Information Science』, 2007, p5.

다) 개념-사례 관계

개념과 사례 간의 관계는 개념을 나타내는 개념과 그 개념의 구체적인 사례가 되는 개체와의 관계를 의미한다. 이때 구체적인 사례는 본질 속성을 가지고 있어야 한다. 여기서 본질 속성은 어떤 것의 아이덴티티를 결정짓는 속성을 말한다. 예를 들어 唐, 宋, 元, 明은 중국의 옛 나라 이름이다. 이들 王朝를 포괄할 수 있는 개념이 王朝라고 할 수 있다. 이때 왕조는 개념이고 唐, 宋, 元, 明은 구체적인 사례에 해당한다. 왕조는 왕조로서의 본질 속성을 가지고 있기 때문에 왕조로 불리는 것이다. 그렇다면 ‘後梁, 後唐, 後晉, 後漢, 後周’의 경우는 어떠한가? 이들 나라는 모두 五代十國 시대의 왕조 이름으로 이들을 포괄할 수 있는 개념은 우선 王朝로 볼 수 있다. 王朝와 王朝 중의 하나인 後梁 등의 관계는 개념과 구체적인 사례의 관계로 볼 수 있다. 그런데 이들 왕조는 五代十國이라는 다른 공통점이 있다. 이때에는 五代十國을 개념으로 보고, 後梁, 後唐 등을 구체적인 사례로 보기는 어렵다. 왜냐하면 五代十國을 구성하고 있는 왕조는 왕조라고 불리는 본질 속성과는 관계없는 다른 이유로 五代十國이 되었기 때문이다. 구체적인 사례는 개별 사례 하나하나가 모두 개념으로 지칭할 수 있어야 한다. 즉 ‘後梁, 後唐, 後晉’ 등은 각각을 모두 왕조라고 지칭할 수 있으나, 이들 개별 국가 하나하나를 五代十國이라고 부를 수는 없다. 五代十國은 어느 하나의 나라만이 존재한다고 해서 五代十國이라는 개념을 충족시킬 수는 없다. 이들 개별 왕조가 모두 합쳐져야 비로소 五代十國이 되는 것이다. 그러므로 五代十國은 개념과 사례의 관계가 아니며, 전체와 이를 구성하고 있는 구성 요소의 관계로 보아야 한다. 다음으로 ‘金希聖, 丁應斗, 金三宅’ 등의 공통점을 살펴보자. 이들은 우선 인명이라는 것 이외에 이들은 中宗甲午式年試에 합격한 합격자라는 공통점을 가지고 있다. 그렇다면 中宗甲午式年試를 무엇으로 보아야 할까? 이는 전체, 개념, 집단 중의 하나인데 전체는 구성원이 유한하며, 金希聖 등은 본질 속성은 사람일 것이다. 그런데 이러한 본질 속성과 과거시험명과 필연적인 연관 관계가 없다. 다만 이들을 모두 합하여 전체를 구성하는 관계를 의미하므로 이 경우에는 전체로 보아야 한다. 전통 시대의 과거합격자 정보, 공신명, 상훈 등이 여기에 해당한다. 여기에는 서명과 서명을 구성하는 목차의 관계도 포함된다. 하나의 집단을 구성하고 있는 人名, 團體名, 組織名 등의 관계는 ‘hasMember(isMemberOf)’를 사용하여 집단과 구성원의 관계로 표현한다.³¹⁹⁾

319) 고영만은 개념과 사례의 관계를 특정 고유명의 인명, 단체명, 조직명을 구성하는 경우에 사용하는 것으로 정의하였으나 본 논문에서는 좀 다르게 파악했다.

표 80. hasInstance(isInstanceOf)

관계명	hasInstance(isInstanceOf)	
범위	정의역	치역
	Thing	Thing
예제	王朝 明德	唐, 清, 明 明德1, 明德2, 明德3...

라) 그 밖의 전체-부분 관계

위에서 설명한 관계 이외에 관계가 있을 수 있다. 'hasPart'는 정의역과 그 일부분인 관계를 의미한다. 그런데 사실 위에서 설명한 대부분의 관계는 일반적인 의미로 모두 전체와 부분과의 관계를 나타내고 있다. 위에서 나오는 관계를 'hasPart'로 표현할 수도 있다. 그렇지만 의미적으로 더 정확한 전달을 위해서 복잡하지만 세분화한 것이다.

표 81. hasPart(isPartOf)

관계명	hasPart(isPartOf)	
범위	정의역	치역
	Thing	Thing
예제	숲	나무

마) 관련된 지식 자원(WebResource)

WebResource는 역사 자료를 이해하는 데 도움이 되는 웹 자원에 관한 정보를 연결시키는 프로퍼티이다. 목적어로 URI 주소를 갖는다. WebResource는 OWL에서 개념적으로 동일한 개체를 사용하는 'owl:sameAs'와 구분해야 한다. 'owl:sameAs'는 온톨로지 개체 정합성 측면에서, 두 개의 서로 다른 온톨로지 개체가 실세계에서 동일한 개체임을 표현할 때 사용한다.³²⁰⁾ 그런데 동일한 개체라는 의미는 간단한 것 같으면서도 쉽지 않은 개념이다. 아래 그림에서 보듯이 국사편찬위원회에서 제공하는 한국사 LOD에서 '이순신'을 조회하면, 그 결과 목록에서 '이순신'과 관련 있는 여러 가지 정보 중에 이순신의 저작물인 '난중일기'로 이동할 수 있다. 이때 '저작물_50932(<http://lod.koreanhistory.or.kr/resources/page/저작물>)

320) 강인수·정한민·이승우·김평·이미경·성원경, 「시맨틱 웹 온톨로지에서의 OWL sameAs 적용」, 『정보과학회논문지 : 소프트웨어 및 응용』, 제34권, 제4호, 2007, 359쪽.

50932)’는 亂中日記의 영인본에 대한 설명이다. 이것과 국립중앙도서관의 난중일기 (<http://lod.nl.go.kr/resource/KMO200547877>)가 같다고 볼 수 있을까? 결론부터 말하자면 같다고 하기는 어려울 것이다. 국립중앙도서관에서 발행한 LOD는 국립중앙도서관에서 소장하고 있는 『난중일기』라는 개체에 대한 설명이지 난중일기라는 하나의 개념에 대해서 설명하는 것은 아니기 때문이다. 예를 들어 국립중앙도서관에서 발행하는 LOD에는 『난중일기』가 여럿이 있을 수 있다. 『난중일기』라는 물리적인 실체가 서로 다르기 때문에 그 책에 대한 설명을 LOD로 발행했기 때문이다. EDM(Europeana Data Model)에서 자원과 자원을 디지털화한 것을 구분한 것도 비슷한 맥락일 것이다.



그림 44. 한국사 LOD 연계 개념

국립중앙도서관에서 발행하는 LOD는 edm:WebResource과 가깝다고 할 수 있다. EDM(Europeana Data Model)에서는 웹 자원에 관한 정보를 담고 있는 클래스(edm:WebResource)를 두었다. EDM에서는 문화 유산 그 자체에 대한 정보와 문화유산에 관한 디지털 웹 자원의 정보를 구분하였다. 예를 들어 실물인 모나리자에 대한 설명을 그 자체에 대한 설명이라 한다면, 모나리자를 디지털화한 사진에 대한 정보는 웹 자원이라고 할 수 있다. edm:WebResource는 내부 혹은 외부 자원과 'dcterms:hasPart, dcterms:isPartOf, owl:sameAs' 등 더블린 코아와 OWL의 어휘를 사용하여 다른 웹 자원과의 관계를 표현하였다. 그런데 EDM에서 말하는 WebResource는 일종의 콘텐츠라고 할 수 있다.

2. LOD 발행 절차

가. LOD 발행 개요

W3C에서 LOD 발행을 위하여 제안한 라이프 사이클은 ‘Specify, Model, Generate, Publish, Exploit’ 등 5단계로 진행되는 절차이다. 그러나 본 논문에서는 자료 구축의 효율성을 고려하여 Modeling, Generate, URI 부여, Store, Publish, Interliking의 6단계로 진행한다. 일반적으로 데이터를 서로 연계할 수 있도록 하기 위해서 팀 버너스 리가 제시한 아래의 4가지 원칙을 충실하게 지킬 수 있도록 한다.³²¹⁾

1. URI를 이용하여 자원을 식별하라.
2. 기계나 사람이 URI로 식별된 자원을 룩업(look up, 탐색, 참조)할 수 있도록 하기 위해 HTTP(프로토콜)를 이용하라.
3. 기계나 사람이 식별된 URI를 룩업했을 때 RDF 또는 SPARQL 등의 표준화된 형식으로 해당 URI가 포함하는 풍부한 정보를 제공하라.
4. 또 다른 자원으로의 연결(link)을 통해 더 많은 지식을 발견하고 활용할 수 있도록 하라.

첫 번째 자원을 URI를 이용하여 식별한다는 것은 데이터 웹의 개체, 개념들에 중복이 되지 않는 유일한 이름을 부여한다는 것을 말한다. 기존 웹에서 우리는 어떤 정보나 서비스에 접근하기 위해서 URL 주소 체계를 이용했다. URL 주소도 URI의 하나이다. 마찬가지로 웹을 플랫폼으로 하는 LOD에서도 웹에서 특정 개체나 개념을 다른 것과 구분 짓고 고유하게 식별하기 위해서 URI를 활용한다.³²²⁾

두 번째 HTTP 프로토콜을 이용해 해당 개념에 접근(look up)할 수 있도록 해야 한다는 것은, 사람들이 HTTP 프로토콜로 개념으로 접근했을 때 서비스를 제공할 수 있는 물리적인 웹 페이지가 제공되어야 한다는 의미이다. OWL에서 URI 주소가 물리적으로 존재하지 않아도 관계가 없으나 LOD를 발행하기 위해서는 서버가 작동하여 요청에 응답할 수 있는 페이지를 구성해야 한다. 예를 들어 DBpedia에서 ‘http://dbpedia.org/page/Seoul’를 이용하여 ‘서울(Seoul)’이라는 객체의 URIs에 접근하면 아래 그림과 같은 웹 페이지를 확인할 수 있다.

321) Linked Data, Tim Berners-Lee, <https://www.w3.org/DesignIssues/LinkedData.html>

322) 한국정보화진흥원, 「알기 쉬운 Linked Open Data」, 2015, 29쪽.

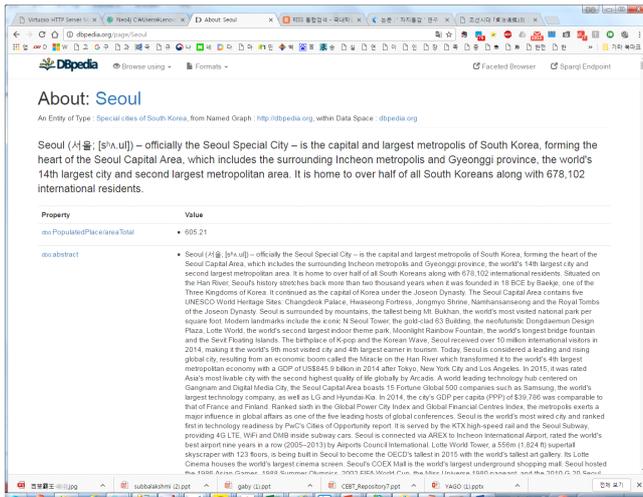


그림 45. DBpedia 서비스

세 번째 RDF 또는 SPARQL 등의 표준화된 형식으로 해당 URI가 포함하는 풍부한 정보를 제공하라는 것은 데이터는 RDF를 사용하고 조회하거나 조회한 결과를 돌려주는 것을 SPARQL을 사용한다는 의미이다. RDF(S), SPARQL은 기존의 웹에서는 사용하지 않던 새로운 기술이다.³²³⁾

네 번째 다른 자원으로서의 연결(link)을 통해 더 많은 지식을 발견하고 활용할 수 있도록 하라는 말은, 기존의 문서의 웹에서도 하이퍼링크를 이용하여 관련있는 정보로 이동할 수 있었던 것처럼 데이터 웹인 LOD에서도 데이터 개체 간의 연결을 통해서 데이터 웹을 보다 풍성하게 만들자는 것이다.³²⁴⁾ 이런 예는 다음 그림과 같이 해당 객체를 정의하고 있는 또 다른 객체로의 연결되는 부분에서 확인할 수 있다.³²⁵⁾

323) 한국정보화진흥원, 「알기 쉬운 Linked Open Data」, 2015, 33쪽.

324) 한국정보화진흥원, 「알기 쉬운 Linked Open Data」, 2015, 34쪽.

325) 박진호, 「도서관 데이터의 링크드 데이터(Linked Data) 변환과 인터링킹(interlinking)을 통한 정보연계 확장성에 관한 연구」, 성균관대학교 일반대학원, 석사학위논문, 2013, 12쪽.

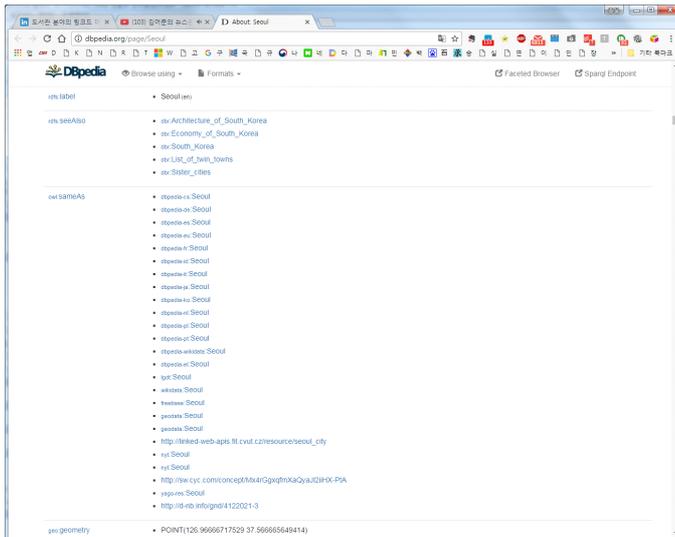


그림 46. DBpedia에서 “서울” 개체와 다른 개체와의 연결 예시

1) Modeling 단계

RDF는 Subject, Predicate, Object의 트리플 형태로 이루어져 있으므로 일반적인 역사서에 나타나는 행위나 사건을 표현하는 데는 취약하다. 이를 해결하기 위한 방법으로 III장에서 역사적 지식을 이벤트와 의미역을 이용하여 표현하는 것으로 설계하였으며, 역사서에서 쉽게 얻을 수 없는 정제된 지식을 수용할 수 있도록 지식 관계망이란 개념을 도입하여 최종적인 모델이 완성되었다. 이러한 모델링에는 Protégé5.2을 이용하여 설계했으며, 기본 언어는 OWL-DL을 사용하였다.

설계가 완료된 온톨로지 모델은 논리적으로 오류가 없는지 검증을 해야한다. 주로 모델링 도구에서 제공하는 추론기(reasoners)³²⁶를 이용한다. 추론기를 활용하여 Ontology에서의 모든 명령문이나 정의가 상호 간에 모순이 없는가를 체크하고 또한 정의에 적합한 개념인지를 확인하였다. protégé에는 Fact++, Hermit, Pellet 등의 추론기가 내장되어 있다. 본 연구에서는 Hermit, Pellet를 기본으로 하여 검증을 하였다. 다음의 예는 hasPosthumousName의 Range의 데이터 형식을 language로 설정했는데, 실제 인스턴스를 입력할 때 언어 설정을 하지 않았다는 메시지를 보여주고 있는 예이다. 해당 부분을 클릭하면 수정해야 하는 부

³²⁶ 추론기에 대한 내용은 ‘<http://owl.cs.manchester.ac.uk/tools/list-of-reasoners>’ 사이트에 잘 정리되어 있다.

분으로 이동한다.³²⁷⁾ 이 단계에서 오류 부분을 수정해 주고 반복적으로 검증을 진행하면 된다.

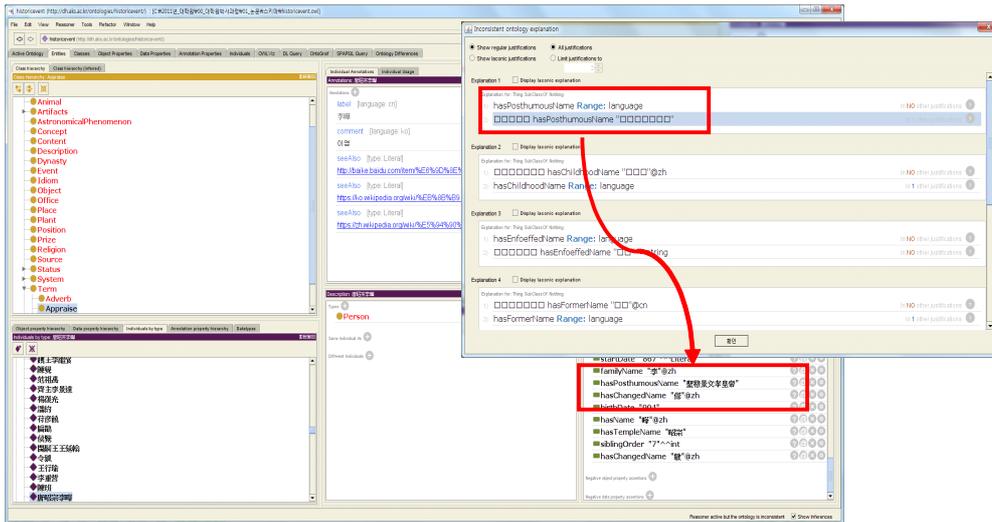


그림 47. protégé Reasoner를 이용한 오류 점검의 예

2) Generate 단계

Model 단계에서 설계된 온톨로지 모델을 바탕으로 실제 데이터에서 지식을 RDF로 변환하는 단계이다. RDB와 같이 구조화된 데이터는 RDB2RDF나 D2R Server³²⁸⁾ 같은 도구를 활용하여 변환한다.³²⁹⁾ D2R Server는 관계형 데이터베이스 내의 콘텐츠를 LOD로 발행할 수 있도록 지원해주는 도구이다. D2R Server는 데이터베이스 콘텐츠를 RDF 데이터로 매핑하기 위해서 고유의 변환 규칙을 적용할 수 있으며 D2RQ 매핑(mapping)을 사용한다.³³⁰⁾

그러나 일반적인 텍스트에서 RDF 트리플을 추출하는 것은 아직까지는 자동으로 할 수가 없다. 데이터에서 RDF를 추출하는 방법은 템플릿을 이용하는 방법과 도구를 활용하는 방법

327) protégé는 영어권에서 개발된 도구라서 그런지 한국어 지원이 완벽하지 않아 화면처럼 글자를 제대로 보여주지 못하는 오류가 있다. 그리고 버전에 따라 기본으로 제공하는 추론기가 달라서 본 논문에서는 protégé4.3과 5.2 버전을 같이 사용하였다.

328) Accessing Relational Databases as Virtual RDF Graphs, <http://d2rq.org>

329) RDB를 RDF 트리플로 변환하는 도구는 W3C의

‘<https://www.w3.org/2001/sw/rdb2rdf/wiki/Implementations>’에 잘 정리되어 있다. 사용자의 환경에 따라 적절한 도구를 선택하면 RDB를 변환할 수 있다.

330) 한상은, 「오픈데이터의 링크드데이터(Linked Data) 발행방법 간 유용성 비교평가에 관한 연구」, 성균관대학교 석사학위논문, 2014, 11쪽.

이 있다. 템플릿을 이용하는 방법은 보통 문서를 직접 보면서 필요한 요소를 템플릿에 입력하고 이를 변환하는 방법이다.³³¹⁾ 이러한 방법은 구축된 RDF를 검증하는 작업이 매우 어렵다는 문제가 있다. 여기에서 좀 더 발전된 방법은 Protégé와 같은 도구를 활용하는 것이다. Protégé는 온톨로지 모델링뿐만 아니라 인스턴스를 추가하는 기능이 있어, 인스턴스를 추가하고 인스턴스 간의 프로퍼티를 연결할 수 있다. 이 방법은 템플릿을 이용하여 RDF 트리플을 만드는 방법에 비해 논리적인 오류를 점검하면서 구축할 수 있다는 장점은 있지만, 속도와 검증이라는 측면에서는 크게 다르지 않다는 단점 또한 있다. 이밖에 전용 도구를 활용하여 구축하는 방법도 있다. 최지예는 말뭉치 부착 도구를 개발하여 데이터 본문을 열람하면서 이벤트를 부착하는 방법을 연구하였다. 이 방법은 말뭉치를 부착하고자 하는 텍스트를 불러온 다음 본문에 출현하는 개체의 위치값을 저장하고 저장된 개체에 존재하는 관계를 연결시키는 방법이다.³³²⁾

그 밖에 Sheffield 대학에서 제공하는 GATE Developer³³³⁾를 이용할 수도 있다. GATE Developer는 자연어에 대한 온톨로지 처리를 목적으로 만들어 졌다. 아래 화면은 중국의 북경대학교에서 CBDB를 구축하면서 GATE라는 온톨로지 구축 도구를 사용하여 관계 정보를 정리하는 화면이다.

331) 정도현 외, 앞의 논문(제13권, 제5호, 2012), 36쪽.

332) 최지예·김명근·박소영, 「문화유산정보 말뭉치 구축을 위한 개체명 및 이벤트 부착 도구」,

『韓國컴퓨터情報學會論文誌』, 2012, 31쪽에 개체명 및 이벤트 부착 도구에 대해서 간단히 설명이 되어 있다.

333) University of Sheffield, <https://gate.ac.uk>

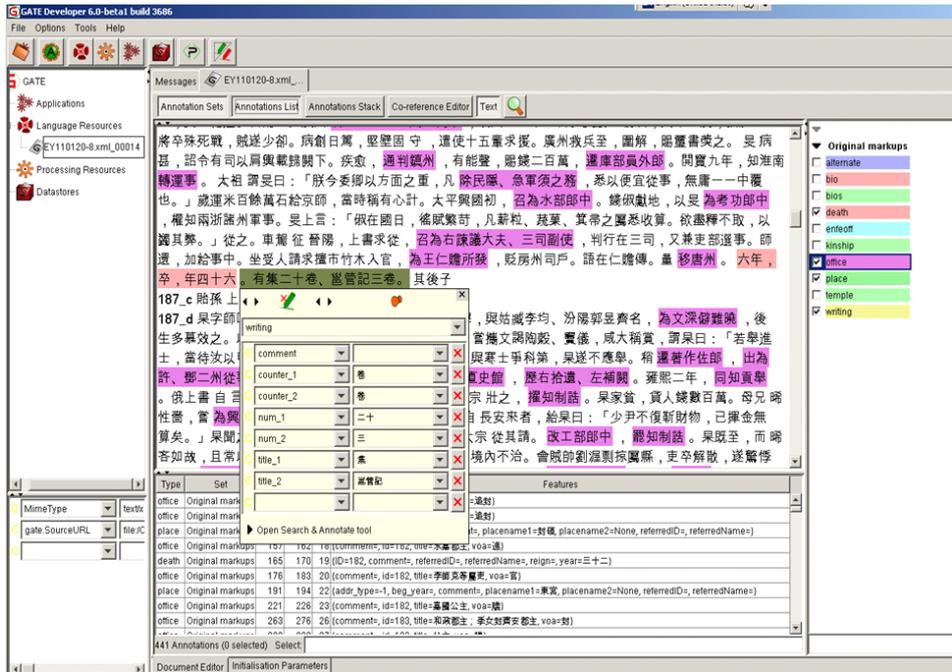


그림 48. Gate Developer를 이용하여 관계를 표현하는 화면

본 논문에서는 검증의 편의성을 위하여 대상 자료인 XML 문서에 필요한 요소를 직접 마크업하고 프로퍼티를 XML 엘리먼트로 마크업한다. 마크업하는 방법은 먼저 본문 중에 나타나는 개체명을 대상으로 마크업한다. 개체명의 time, person, place, event 등이며 개체명의 종류는 고정되어 있지 않고 증가할 수 있다. 이 단계가 끝나면 XML 문서에서 마크업된 요소를 SQL에 인서트한다. 이 단계를 하기 위해서는 별도의 프로그램이 필요하다. 다음으로는 XML 문서에서 마크업된 요소를 중심으로 Protégé에 직접 인스턴스를 등록한다. Protégé에서 인스턴스를 등록하고 필요한 관계를 Object Property와 Datatype Property로 구분해서 연결해 준다. Protégé는 복잡한 온톨로지를 구조를 직접 입력하지 않아도 자동으로 OWL을 생성해 주기 때문에 편리하다. 그 다음에는 Protégé에서 저장된 OWL 파일을 열어 입력된 부분을 찾아서 OWL 문서의 구조를 파악한다. 그리고 그 구조에 맞춰서 MS-SQL에서 쿼리를 작성한다. 이 쿼리를 이용하여 OWL 문서로 변환한 다음 이 문서를 OWL 문서에 붙여 넣고 저장한다. 그러면 Protégé에서 다시 읽어들이지를 묻는데 확인을 누르면 된다. 같은 이름의 자원이 생성이 되면 자동으로 병합하여 하나로 만들어 준다. 그러므로 자원이 다른 것은 반드시 식별하여 구분해 주어야 한다.

표 82. SQL 쿼리를 이용한 RDF 트리플 변환 예

```
SELECT '<owl:NamedIndividual rdf:about="http://www.history.com/yoons#event'+번호+'">
  <rdf:type rdf:resource="&kno;War"/>
  <kno:after rdf:resource="http://www.history.com/yoons#event'+번호+'"/>
  <kno:actor rdf:resource="http://www.history.com/yoons#+A+'"/>
  <kno:patient rdf:resource="http://www.history.com/yoons#+PA+'"/>
  <kno:action rdf:resource="http://www.history.com/yoons#+기본형+'"/>
  </owl:NamedIndividual>'
FROM 통감절요_고유명사RDF
WHERE A !=''
```

XML 문서 내에 사용하는 태그의 표지는 뒤에서 상세하게 다룬다. 추출된 RDF 트리플은 Protégé로 작성한 OWL 문서에 삽입한다.

3) URI 부여

LOD 발행의 첫 번째 원칙이 바로 URI를 이용하여 자원을 식별하라는 것이었다. URI 구성 체계는 사람이 이해할 수 있는 분류 체계로 구성하는 것이 좋다.³³⁴⁾ 또한 식별자가 무엇을 설명하는지 반영하는 것이 좋다.

일반적으로 RDB를 온톨로지로 변환할 때는 RDB에 부여된 ID 등을 이용하여 URI를 발행하기 때문에 특별한 어려움을 없으나, 역사 자료에 나오는 인물을 대상으로 URI를 발행하기 위해서는 이들을 모두 정리하고 고유한 식별자를 부여해야 한다. 앞서 人名과 地名을 대표명칭으로 정리했기 때문에 이 대표명칭을 활용하면 URI가 중복해서 발행되는 것을 막을 수 있다.

4) Store 단계

Store 단계는 Generate 단계에서 생성된 RDF 트리플을 데이터베이스에 저장하는 단계를 말한다. 일반적으로 데이터베이스는 관계형 데이터베이스와 NoSQL(Not Only SQL)로 구분할 수 있는데 RDF 데이터베이스는 NoSQL의 하나이다. RDF 트리플을 저장할 수 있는 데이터베이스는 다양하다.³³⁵⁾ 다음의 표는 일반적으로 사용하는 RDF 트리플 저장소를 정리한

334) David Wood·Marsha Zaidman·Luke Ruth(저)/오원석·이명진·박진호·김보람(역), 『Linked Data - Structured data on the web』, 2014, 32쪽.

335) 한국문화정보센터, 「지능형 웹 기반 문화정보 활용 활성화 연구 보고서」, 2011, 31쪽.

것이다.

표 83. RDF 데이터베이스³³⁶⁾

Name	URL
OntoBase	http://li-st.com/jsp/product/ontobase.jsp
4Store	http://www.4store.org/
Allegro-Graph	http://www.franz.com/agraph/allegrograph
BigData	http://www.bigdata.com/
Fuseki	http://jena.apache.org/documentation/serving_data/index.html
Mulgara	http://mulgara.org/
Oracle	http://www.oracle.com/technetwork/database/options/semantic-tech/index.html
OWLIM	http://www.ontotext.com/owlim
Redland RDF Library	http://librdf.org/
Sesame	http://www.openrdf.org/
StarDog	http://stardog.com/
Virtuoso	http://virtuoso.openlinksw.com/

본 논문에서는 'Virtuoso RDF Store'를 사용하여 RDF 트리플을 저장하였다. Virtuoso RDF store는 다양한 RDBMS와 ODBMS, 텍스트, HTML, XML 데이터를 하나로 통합하는 미들웨어로, 다양한 형식의 데이터를 RDF 형식으로 변환하여 RDF 트리플스토어에 저장한다. 또한 SPARQL Endpoint 제공을 통해 HTML 형식으로 RDF 데이터 접근을 제공하는 Linked Data의 발행, 저장, 접근 및 관리 기능을 제공한다. LOD 발행은 작성한 온톨로지 업로드, 가상 디렉토리 생성, URL Rewring 규칙 작성 등의 절차로 진행된다.³³⁷⁾

RDF 데이터베이스에 대한 비교 설명은 「지능형 웹 기반 문화정보 활용 활성화 연구 보고서」에

상세히 정리되어 있으므로 이 자료를 참조하면 좋다.

336) David Wood 외(저)/오원석 외(역), 앞의 책(2014), 180쪽.

337) 이명진, 「링크드 오픈 데이터 -소개부터 구축 및 활용까지」, 리스트, 78~91쪽.

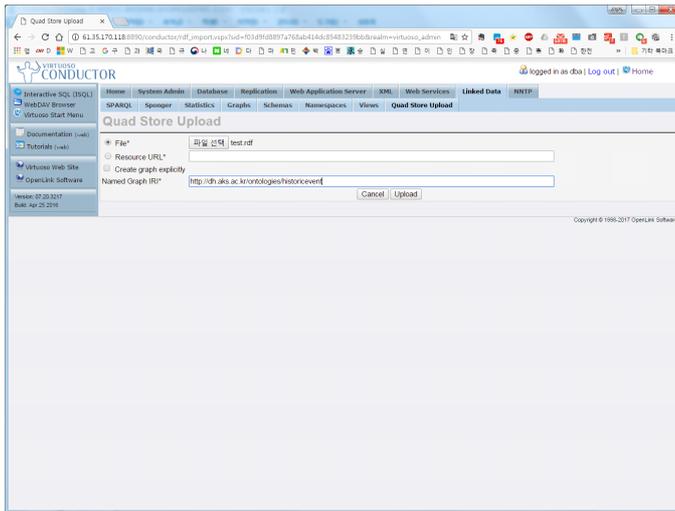


그림 49. Virtuoso를 이용하여 RDF 트리플을 Store하는 단계

5) Publish 단계

Publish 단계는 RDF 데이터베이스에 저장된 RDF 트리플을 웹으로 발행(Publish)하여 외부에서 데이터를 사용할 수 있도록 하는 단계이다. Linked Data를 외부로 공개하고 HTTP를 통해 유통하기 위해서는 외부의 응용 서비스나 데이터와의 점점 제공을 위한 SPARQL Endpoint가 필요하다. SPARQL Endpoint는 발행된 Linked Data에 대해 사용자가 SPARQL을 사용하여 원하는 결과를 반환할 수 있도록 지원하고, 외부 Linked Data에서 링크를 통해 특정 자원의 정보를 요청할 때 이를 반환해 주는 역할을 수행한다.³³⁸⁾

표 84. LOD 발행 도구

Name	URL
Pubby	http://wifo5-03.informatik.uni-mannheim.de/pubby/
Virtuoso	http://virtuoso.openlinksw.com/
BigData	http://www.bigdata.com/
Fuseki	http://jena.apache.org/documentation/serving_data/index.html

Virtuoso는 정적 HTML 페이지 및 Virtuoso Server Pages(VSP)를 통해 동적 콘텐츠를 위해서 HTTP 웹서버를 통합하고 있다. 또한 웹 서비스 생성 및 호스팅을 지원한다.³³⁹⁾

338) 한국문화정보센터, 앞의 책(2011), 29쪽.

339) 한국문화정보센터, 위의 책(2011), 32쪽.

이 단계를 통해서 RDF 트리플 형태로 공개되며 이 시맨틱 웹 표준인 RDF 형태의 데이터로 발행(publish)되므로 마치 하나의 글로벌 데이터베이스처럼 질의하고 이용이 가능하다.

6) 인터링킹(Inter-linking) 단계

데이터 연계를 위해서는 데이터 셋 간에 동일 자원에 할당된 서로 다른 URI(Uniformed Resource Identifier)를 찾아서 연결하거나, 기존에 할당된 URI를 그대로 활용하는 것이 매우 중요하다. 외부에 존재하는 동일한 개념을 나타내는 자원과 연결하는 것을 인터링킹(Interlinking)이라고 한다. 인터링킹은 데이터베이스 내부에서의 다른 데이터들끼리 연결하는 것뿐 아니라 외부자원 상호 간에도 연결이 가능하다.³⁴⁰⁾ 이를 통해서 잠재적 지식을 발견하고, 관련 있는 지식을 확장시킬 수도 있다. 또한 인터링킹을 통해서 자원의 중복 구축을 미연에 방지할 수도 있다. 인터링킹은 크게 두 가지로 구분할 수 있다. 먼저 'owl:sameAs'를 사용하여 의미적으로 동일한 외부의 자원을 연결하는 것과 'rdfs:seeAlso'를 사용하여 자원에 대한 추가적인 정보를 제공하는 것이 바로 그것이다.³⁴¹⁾ 'owl:sameAs'는 분산된 데이터 세트와 식별된 정보 자원을 링크드 데이터로 통합한다. 만약 기술하고 있는 자원이 'owl:sameAs'로 연결되어 있다면, 컴퓨터는 해당 두 자원을 같은 자원으로 판단해 두 자원 사이의 다른 속성값을 합칠 수 있도록 한다. 'rdfs:seeAlso'에서 추가적인 정보라고 하는 것은 외부에 있는 같은 수준의 자원을 참조할 수 있게 연결하는 것을 말한다. 이는 텀버너스 리의 다른 URI 링크를 포함하라는 링크드 데이터 4원칙 중 네 번째를 지원해 결과적으로 데이터 발견을 향상시킬 수 있다.³⁴²⁾

LOD에서 owl:sameAs는 아래와 같은 형식으로 표현된다.

```
<http://dbpedia.org/resource/Amsterdam>  
owl:sameAs <http://rdf.freebase.com/ns/...> ;  
owl:sameAs <http://sws.geonames.org/2759793> ;  
...
```

인터링킹의 가장 단순한 방법은 자원의 양이 많지 않을 경우에 관련된 콘텐츠를 사람의 눈으로 직접 식별하면서 데이터에 링크를 추가하는 방식이다. 그러나 이 방법은 정확성은 높으나 많은 시간과 비용이 소요된다는 단점이 있다. 이런 이유로 해서 Openlink Data Explore와 같은 시맨틱 색인을 사용하거나 자동화된 도구를 활용하기도 한다.

340) 이현정·남영준, 「국내 Linked Open Data 발행 후 활용에 관한 연구: 국가서지와 Riss Linked Data를 중심으로」, 『한국정보관리학회 학술대회논문집』, 2014, 112~113쪽.

341) 이명진, 「링크드 오픈 데이터 -소개부터 구축 및 활용까지」, 73쪽.

342) David Wood 외(저)/오원석 외(역), 앞의 책(2014), 226쪽.

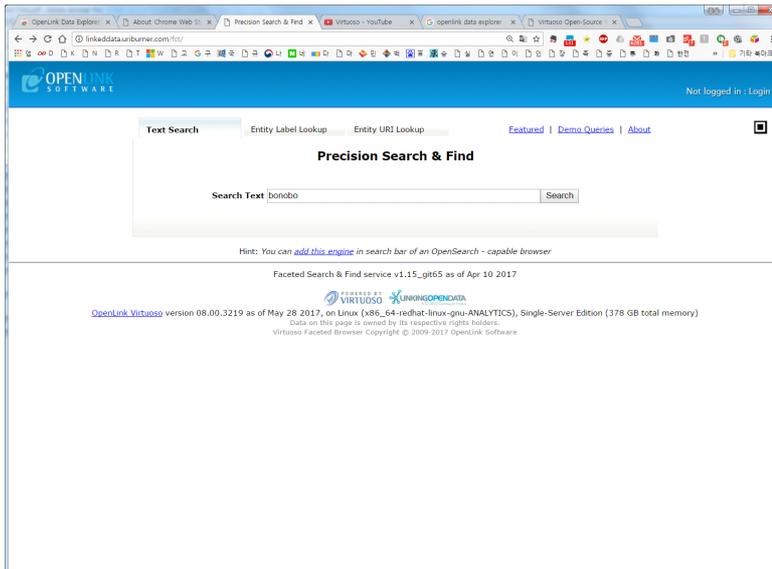


그림 50. Openlink Data Explorer³⁴³⁾

자동화된 도구로는 LIMES³⁴⁴⁾와 SILK³⁴⁵⁾가 있다. 이들 도구는 자동으로 소스가 되는 데이터 셋에서 의미있는 인스턴스를 추출하고, 대상이 되는 데이터 셋으로부터 최대한 유사한 인스턴스를 찾아 추천해 주는 역할을 한다.³⁴⁶⁾ 두 데이터 셋의 연결을 위해서 두 데이터 셋 사이의 연결 규칙을 지정하고, 이 연결 규칙을 바탕으로 연결할 자원을 찾아 두 데이터 셋이 연결될 수 있도록 제공한다. LIMES는 이러한 연결 규칙을 지정하기 위해 Link 명세서를 작성해야 하며, Link 명세서는 XML 기반으로 개발된 LIMES Configuration Language라는 언어를 통해 작성해야 한다.³⁴⁷⁾ 사용자가 LIMES를 사용하기 위해서는 새로운 언어를 습득해야 하는 단점이 있다. 반면에 SILK 사용자가 서로 다른 데이터 셋의 연결 작업을 쉽게 수행할 수 있도록 SILK Workbench³⁴⁸⁾라는 웹 응용 프로그램을 제공한다. LIMES와 SILK는 아직까지 신뢰성을 담보할 수 없으므로 본 논문에서는 수작업에 의해 인터링킹을 정보를 구축하였다.

343) OPENLINK SOFTWARE, <http://linkeddata.uriburner.com/fct>

344) LIMES, AKSW, <http://aksw.org/Projects/LIMES.html>

345) SILK, The Linked Data Integration Framework, <http://silkframework.org>

346) 이경욱, 「LOD InterLinking」, 다이퀘스트, 2015, 27쪽.

347) 강민아, 「NTIS 연구정보 개방을 위한 LOD 구축 및 활용 방법 구현」, 충남대학교 대학원 석사학위논문, 2015, 39쪽.

348) SILK, The Linked Data Integration Framework, <http://silkframework.org>

나. LOD 구축 과정

1) Generate(인스턴스의 생성)

인스턴스를 생성하기 위해서는 먼저 온톨로지 모델이 있어야 한다. 이 모델을 만들기 위해서 먼저 본문에 출현하는 개체명을 일일이 마크업하였다. 마크업하면서 처음 정의한 개체명 이외에도 추가적으로 의미가 있는 요소가 발견된 때마다 추가로 마크업하였다. 마크업은 아직까지 자동 태깅 기술이 원하는 수준까지 발전되지 않았기 때문에 EM-EDITOR³⁴⁹⁾에서 직접 개체에 대해서 매크로 기능³⁵⁰⁾을 이용하여 태깅을 하였다. 태깅이 끝나면 태깅된 요소를 대표명칭으로 정리하여 동일한 사람이나 지명을 지칭하는 명칭을 정리하였다. 태깅과 온톨로지 설계는 거의 동시에 진행이 되며, 태깅하면서 새로운 요소가 발견되면 온톨로지 설계에 반영을 하였다. 기본적인 온톨로지 설계가 끝나면 실제로 태깅된 데이터를 직접 인스턴스로 등록하면서 모델의 유효성을 확인한다.

역사 자료의 인스턴스 생성은 템플릿을 이용하는 방법과 도구를 이용하는 방법이 있는데, 본 논문에서는 XML 문서에 직접 관계를 속성으로 표현하고 마크업된 요소를 태그 추출 프로그램을 이용하여 SQL DB에 저장한 후, 이를 다시 DB에서 쿼리를 이용하여 원하는 형태의 RDF 트리플로 생성하는 방법을 사용하였다. 또한 『통감절요』에 출현하는 개체를 참조할 수 있도록 지식 연계를 병행하였다.

가) 지식 연계

위에서 언급한 인터링킹과 비교하자면, 인터링킹이 LOD로 발행된 자료가 연결하는 것이라면 지식 연계는 아직 LOD로 발행되지 않은 자료를 연결하는 것을 의미한다. 예를 들어 中文 위키피디아³⁵¹⁾나 바이두³⁵²⁾, 한국 위키피디아 등이 그 대상이 된다. 중문 위키피디아는 『通鑑節要』를 이해하는 필요한 다양한 정보를 제공하고 있다. 『通鑑節要』 중 오대십국

349) emeditor은 아마도 현재까지 나온 일반 텍스트 에디터 중에서 가장 뛰어난 성능을 가진 에디터일 것이다. 최대 248기가의 문서까지 편집이 가능하며, 안전한 Unicode 편집의 지원, 다양한 플러그인 기능, 안정성 있는 문서 편집 등은 세계 최고의 에디터라고 부리기에 손색이 없다.(<https://ko.emeditor.com>) 본 연구에서는 emeditor을 이용하여 10gb가 넘는 wikidata 데이터를 열어서 구조를 확인할 수 있었다.

350) emeditor에서는 뛰어난 매크로 기능을 제공한다. 본 논문에서는 태깅을 할 때 블록을 잡고 단축키를 누르면 블록으로 설정된 부분이 자동으로 태깅되게 하는 매크로를 만들어 메뉴와 단축키로 등록한 다음 해당 기능을 이용하여 태깅을 하였다.

351) 維基百科, <https://zh.wikipedia.org>

352) baidu 百科, <https://baike.baidu.com>

시대에 출현하는 전체 594명의 인물 중 445명의 인물이 중문 위키피디아와 연결이 되므로 적지 않은 인물이 연결되고 있다고 할 수 있다. 위키피디아와 연결 작업은 『通鑑節要』 본문을 읽으면서 인물의 행적과 위키피디아의 설명을 비교하여 일치하는지 여부를 확인하면서 일일이 정리하였다.³⁵³⁾ 지식 연계의 OWL 문서에서 다음과 같이 정리된다.

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/Book/kno#孟子 -->
<owl:NamedIndividual rdf:about="&knoBook;孟子">
  <rdf:type rdf:resource="&kno;Book"/>
  <rdfs:label xml:lang="cn">孟子</rdfs:label>
  <rdfs:label xml:lang="kr">맹자</rdfs:label>
  <rdfs:seeAlso rdf:datatype="&rdfs:Literal">http://baike.baidu.com/item/%E5%AD%9F%E5%A
D%90/10855864#viewPageContent</rdfs:seeAlso>
  <rdfs:seeAlso rdf:datatype="&rdfs:Literal">http://baike.baidu.com/item/%E5%AD%9F%E5%A
D%90/126</rdfs:seeAlso>
  <rdfs:seeAlso rdf:datatype="&rdfs:Literal">https://ko.wikipedia.org/wiki/%EB%A7%B9%EC%9
E%90_(%EC%B1%85)</rdfs:seeAlso>
  <rdfs:seeAlso rdf:datatype="&rdfs:Literal">https://zh.wikipedia.org/wiki/%E5%AD%9F%E5%A
D%90</rdfs:seeAlso>
</owl:NamedIndividual>
```

아래는 본 연구를 통해서 owl:seeAlso를 통해서 참조할 수 있도록 연계한 결과이다.

표 85. 지식 연계 자료량

개체명	종수	연결 결과
인물 연계	569	한글 위키(147), 중문 위키(445), 바이두(398)
지명 연계	369	한글 위키(61), 중문 위키(281), 바이두(277), CHGIS(234)
관직명 연계	305	한글 위키(21), 중문 위키(122), 바이두(138)
관서명 연계	12	중문 위키(3), 바이두(10)
종족 연계	19	중문 위키(1)
서명 연계	44	한글 위키(21), 중문 위키(35), 바이두(33)
왕조 연계	101	중문 위키(99)

나) event의 표현

OWL 문서에 필요한 RDF 트리플을 생성하기 위해서 본문 중에 나타나는 관계를 이벤트(event) 엘리먼트의 속성으로 표현한다. 이 말은 XML 문서 본문 중에 행위를 나타내는 동사를 event 태깅을 하고, event 엘리먼트에 앞서 언급한 의미역의 구분에 따라 속성을 넣은 다음 해당 속성에 필요한 정보를 앞뒤 내용을 읽으면서 직접 속성값으로 넣는다는 것을 말한다. 이때 속성은 앞서 정의한 意味役의 약칭으로 부여한다. 아래 박스처럼 event 엘리먼트 내에서 동사는 VV, 행동주는 AG, 피동주는 PA, 시간은 TM 등의 약칭으로 속성을 만든

353) http://dhlab.aks.ac.kr/~yju2999/wiki/index.php/통감절요_출현_인물

다음 앞 뒤 문맥을 보면서 각 개별값을 채워준다.

표 86. XML 문서 내에 RDF 트리플의 표현

```
<time>冬十月</time>에 <person>行言</person>이 <event ref="16" DA="854.10" VV="제수하다02" GO="海州刺史" AG="唐僖宗李儂" year="854" PA="李行言" month="10" event="event_16">除</event><position>海州刺史</position>하니 <event ref="17" VV="입조하다01" AG="李行言" event="event_17">入</event><event ref="18" VV="사은하다" AG="李行言" PA="唐僖宗李儂" event="event_18">謝</event>어늘 上이 <event ref="19" VV="하사하다01" event="event_19">賜</event>之<thing name="紫金魚袋">金紫</thing>하고 問曰 卿이 知所以衣紫乎아 對曰 不知이다 上이 命取殿柱之帖하야 示之하다
```

이 과정은 본문 중에 많은 마크업이 부가되었기에 문서의 가독성이 떨어지기 때문에 매우 정밀한 작업이 요구된다. event의 대상이 명사구나 명사절이 되는 경우도 적지 않다. 아래 문장에서 韓全誨가 요청한 것은 명사구 ‘鳳翔으로 갈 것’이다.

```
<time>十一月</time>에 <position>中尉</position><person>韓全誨</person>等이 <event ref="449" TM="901" VV="진열하다01" TE="兵" AG="韓全誨" year="901" LO="殿前" event="event_449">陳</event>兵殿前하고 <event ref="451" TM="901" VV="청하다" AG="韓全誨" year="901" event="event_450" TE="event_451" PA="唐昭宗李晔">請</event><event ref="450" TM="901" VV="가다01" AG="唐昭宗李晔" year="901" event="event_451" GO="鳳翔">幸</event><place>鳳翔</place>이어늘
```

위의 문장에서 ‘韓全誨’가 청한 것은 event_451인 ‘幸鳳翔(鳳翔으로 가다)’가 된다. 이때 ‘가다’의 주어는 ‘唐昭宗李晔’이 목적지가 鳳翔되며 이 행위는 구체적으로 실현이 된 것이 아니다. 이처럼 구체적으로 실현된 것이 아닌 의도의 대상이 되는 행위는 구분해 주어야 한다. 그 이유는 본 연구에서 모든 행위를 event로 표현을 하는데 구체적으로 발생한 행위와 의도의 대상이 되는 행위를 구분하지 않으면 SPARQL에서 조회할 때 실제로 발생한 행위와 어떤 의도에서의 행위를 구분할 수 없기 때문이다. 모든 event는 출현한 순서에 따라 일련번호를 부여하여 ID로 삼는다.

다) Relation의 표현

역사서에는 event와 같은 동적인 관계뿐만 아니라 사람과 사람의 관계, 사람과 관직의 관계 등 잘 변하지 않는 관계도 있다. 다음은 이들을 표현하기 위한 방법이다. 사람과 사람의 관계는편의상 국역본을 대상으로 작성한다.

```
<position>翰林學士</position> <person type="성명" name="李德裕" P="position" O="翰林學士">李德裕</person>는 <person type="성명" name="李吉甫" P="명" O="吉甫">吉甫</person>의 <relation S="李吉甫" P="子" O="李德裕">아들</relation>이다.
```

사람과 관직의 관계는 ‘person’ 엘리먼트에 P, O 속성으로 관계를 표현한다. 위의 박스에서 ‘name="李德裕" ref="" P="position" O="翰林學士"'는 RDF 트리플에서 ‘李德裕의 관직은翰林學士이다’라는 것으로 표현된다.

가족 관계의 표현도 중요하다. 가족 관계나 사회적 관계 등은 모두 관계명칭을 ‘relation’으로 마크업하고 ‘relation’ 엘리먼트에 S, P, O 속성으로 관계를 표시해 준다.

```
<person type="" pid="" name="唐穆宗李恆" ref="" P="명" O="">穆宗</person>은 이름이 <person type="명" pid="" name="唐穆宗李恆" ref="" P="명" O="39887">恆</person>이니, <person type="" pid="" name="" ref="" P="" O="唐穆宗李恆">憲宗</person>의 <relation S="唐憲宗" P="二子" O="唐穆宗李恆">둘째 아들</relation>이다.
```

간혹 주어와 목적어가 두 개 이상인 경우가 있는데 이때에는 개체와 개체를 ,로 구분해 준다. 복수의 개체는 SQL 쿼리에서 해결한다.

라) 중국 연대 대조표의 작성

『通鑑節要』에서 시간 정보를 표현하기 위해서는 연표를 참조로 해야 한다. 그런데 五胡十六國 시대나 五代十國 시대는 동시에 여러 왕조가 존립하기 때문에 연도 확인이 매우 어렵다. 이를 해결하기 위해 중국의 모든 왕조가 포함된 연표를 작성하였다. 年表의 세로축에는 서기년을 위로 기원전 2697년부터 아래로 2303년까지 이어지게 배열하였으며, 가로축에는 101개의 왕조를 배열하였다. 이 왕조는 중국에 존재했던 대부분의 왕조이다. 이 연대 대조표와 『通鑑節要』에 간지와 王朝名, 年號를 적절하게 이용하여 서기년을 확인한다.

시간 정보를 확인하는데 있어 가장 중요한 점은 『通鑑節要』가 『資治通鑑』을 節錄하면서 여러 날, 심지어는 몇 달의 간격이 있는 기사도 단 한줄로 요약한 경우가 있다는 것이다.³⁵⁴⁾ 본 논문에서는 시간의 표현을 기본적으로 서기 연도로만 표시하였다. 王朝名에는 본 논문에서 정리한 대표명칭 규칙에 따라 모든 황제의 명칭이 정리되었으며 황제가 바뀌는 경우 年號를 /를 이용하여 나란히 기재해 주었다. 대표명칭은 中國皇帝子孫과 中國歷代后妃 테이블에도 모두 동일한 대표명칭으로 정리하여 복잡한 중국의 황실의 가계를 파악할 수 있게 하

354) 이 내용은 II장의 1절 ‘『資治通鑑』 및 『通鑑節要』에서 상세하게 다루었다.

였다.

연번	기원년	간지	대표왕조	南年號年	南漢	南漢年號年	南唐	南唐年號年	後蜀	後蜀年號年	南唐	南唐年號年	北
1	2697	甲子	廣帝野輪氏										
2	2698	乙丑	廣平新輪氏										
3636	938	戊戌	五代	通文3	高祖(劉龔)	大有11	南平文獻王高從誨	後蜀隱帝孟昶	廣政4	南唐烈祖李昇	昇元2		
3637	939	己亥	五代	通文4/永隆	高祖(劉龔)	大有12	南平文獻王高從誨	後蜀隱帝孟昶	廣政5	南唐烈祖李昇	昇元3		
3638	940	庚子	五代	永隆2	高祖(劉龔)	大有13	南平文獻王高從誨	後蜀隱帝孟昶	廣政6	南唐烈祖李昇	昇元4		
3639	941	辛丑	五代	永隆3	高祖(劉龔)	大有14	南平文獻王高從誨	後蜀隱帝孟昶	廣政7	南唐烈祖李昇	昇元5		
3640	942	壬寅	五代	永隆4	高祖(劉龔)/海陵	大有15/光	南平文獻王高從誨	後蜀隱帝孟昶	廣政8	南唐烈祖李昇	昇元6		
3641	943	癸卯	五代	永隆5/天德	高祖(劉龔)/海陵	大有16/廣	南平文獻王高從誨	後蜀隱帝孟昶	廣政9	南唐烈祖李昇	昇元7/保大1		
3642	944	甲辰	五代	天德2	中宗文武光聖明	乾和2	南平文獻王高從誨	後蜀隱帝孟昶	廣政10	南唐元宗李璟	保大2		
3643	945	乙巳	五代	天德3	中宗文武光聖明	乾和3	南平文獻王高從誨	後蜀隱帝孟昶	廣政11	南唐元宗李璟	保大3		
3644	946	丙午	五代		中宗文武光聖明	乾和4	南平文獻王高從誨	後蜀隱帝孟昶	廣政12	南唐元宗李璟	保大4		
3645	947	丁未	五代		中宗文武光聖明	乾和5	南平文獻王高從誨	後蜀隱帝孟昶	廣政13	南唐元宗李璟	保大5		
3646	948	戊申	五代		中宗文武光聖明	乾和6	南平文獻王高從誨/高保融	後蜀隱帝孟昶	廣政14	南唐元宗李璟	保大6		
3647	949	己酉	五代		中宗文武光聖明	乾和7	高保融	後蜀隱帝孟昶	廣政15	南唐元宗李璟	保大7		
3648	950	庚戌	五代	馬希崇	中宗文武光聖明	乾和8	高保融	後蜀隱帝孟昶	廣政16	南唐元宗李璟	保大8		
3649	951	辛亥	五代		中宗文武光聖明	乾和9	高保融	後蜀隱帝孟昶	廣政17	南唐元宗李璟	保大9	北漢世祖	
3650	952	壬子	五代		中宗文武光聖明	乾和10	高保融	後蜀隱帝孟昶	廣政18	南唐元宗李璟	保大10	北漢世祖	
3651	953	癸丑	五代		中宗文武光聖明	乾和11	高保融	後蜀隱帝孟昶	廣政19	南唐元宗李璟	保大11	北漢世祖	
3652	954	甲寅	五代		中宗文武光聖明	乾和12	高保融	後蜀隱帝孟昶	廣政20	南唐元宗李璟	保大12	北漢世祖	
3653	955	乙卯	五代		中宗文武光聖明	乾和13	高保融	後蜀隱帝孟昶	廣政21	南唐元宗李璟	保大13	北漢世宗	
3654	956	丙辰	五代		中宗文武光聖明	乾和14	高保融	後蜀隱帝孟昶	廣政22	南唐元宗李璟	保大14	北漢世宗	
3655	957	丁巳	五代		中宗文武光聖明	乾和15	高保融	後蜀隱帝孟昶	廣政23	南唐元宗李璟	保大15	北漢世宗	
3656	958	戊午	五代		中宗文武光聖明	乾和16/大寶	高保融	後蜀隱帝孟昶	廣政24	南唐元宗李璟	中興1/交泰	北漢世宗	
3657	959	己未	五代		南漢後主劉鋹	大寶2	高保融	後蜀隱帝孟昶	廣政25	南唐元宗李璟	顯德2	北漢世宗	
3658	960	庚申	宋(北宋)		南漢後主劉鋹	大寶3	高保融/高保勳	後蜀隱帝孟昶	廣政26	南唐元宗李璟	顯德3	北漢世宗	
3659	961	辛酉	宋(北宋)		南漢後主劉鋹	大寶4	高保勳	後蜀隱帝孟昶	廣政27	南唐元宗李璟	顯德4	北漢世宗	
3660	962	壬戌	宋(北宋)		南漢後主劉鋹	大寶5	高保勳/高繼沖	後蜀隱帝孟昶	廣政28	南唐李煜	顯德5	北漢世宗	
3661	963	癸亥	宋(北宋)		南漢後主劉鋹	大寶6	高繼沖	後蜀隱帝孟昶	廣政29	南唐李煜	顯德6/乾道1	北漢世宗	

그림 51. 중국 연대 대조표

이 연표는 지식 관계망에서 OWL의 ‘owl:sameAs’ 속성을 이용하여 年號와 서기년이 연계될 수 있도록 매핑 작업을 한다. 이 작업을 통해서 서기년을 중심으로 왕조, 연호, 묘호 등이 연계되어 시간 관련된 거의 모든 정보가 상호 변환이 가능해 진다.

마) 시간 정보의 표현

다음은 『通鑑節要』에 나오는 시간 정보를 정리하는 방법에 대해서 기술한다. 『通鑑節要』의 가능 큰 특징 중의 하나는 編年體로 歷史가 기술되었다는 점이다. 즉 본문 중에 나타나는 행위는 시간이란 개체와 연결이 되어야 한다는 것이다. 이 말은 『通鑑節要』에 나오는 모든 행위에 대해서 시간을 파악해야 한다는 것이다. 다행스럽게도 『譯註 通鑑節要』는 편집 체제 상 간지와 연도를 기사 앞 부분에 배치하였다. 이 정보와 사진 등을 참조로 하여 event로 마크업된 동사에 시간을 확인하고 서기연도를 TM 속성에 정리해 준다. 편년체 역사서의 특징 상 모든 기사는 연도에 따라 기술되는 것이 원칙이다. 본문 중에는 간혹 春, 夏 등의 계절과 월을 기록하기도 하며, 드물게는 구체적인 날짜까지 기술하기도 하나 이들을 반영하지는 않았다.

바) 의미역의 마크업 검증

다음으로 검증이 필요하다. 검증은 작업의 효율성을 위해 XML 문서에 XSL을 적용하여 화면상 혹은 출력물 형태로 검증을 진행한다. 이때 중요한 것은 본문 중에 개체명으로 표현해야 할 것이 많기 때문에 색상으로 표현하더라도 쉽게 구분하기 어렵기때문에 이를 개선해야 한다는 점이다. 본 논문에서는 이점을 보완하여 색상과 줄, 부호 등을 적절하게 활용하여 event와 개별 개체명을 쉽게 식별할 수 있도록 하였다.

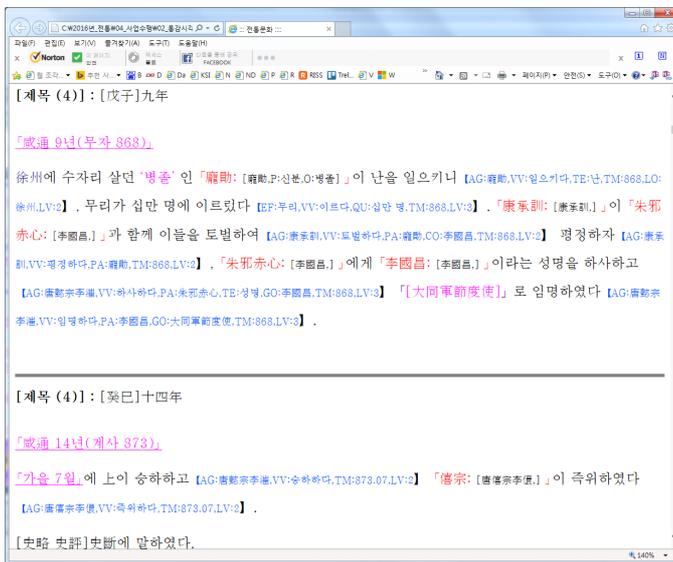


그림 52. XSL을 적용하여 관계를 검증하는 화면

사) XML 데이터에서 태그 추출

XML 문서에 마크업한 요소는 MSSQL로 저장해야 한다. 이때 사용한 도구는 XMLtagrefiner이란 프로그램과 MS-Server2014이다. XMLtagrefiner는 XML의 특성을 활용하여 이용자가 원하는 엘리먼트를 Xpath를 이용하여 지정하고 추출하면 자동으로 해당 엘리먼트를 SQL에 저장해 주는 도구이다.



그림 53. XML 태그 추출기를 이용하여 XML 문서에서 SQL DB로 저장하는 화면

아) 온톨로지 변환

다음으로는 마크업된 데이터를 Protégé에 인스턴스로 추가한다. 우선 RDF 트리플로 변환하기 위해서는 RDF 문서의 구조를 파악해야 한다. 이를 파악하기 위해서는 Protégé에 실제로 인스턴스를 추가한 후 일반 에디터에서 OWL 문서를 열어서 그 구조를 파악해야 한다. 일단 구조를 파악하면 이 구조에 맞춰 태깅한 XML 문서에서 필요한 요소를 추출한다. 그후에는 RDF 트리플 구조에 맞춰 SQL 쿼리를 생성한다. 정형화된 데이터를 RDF 트리플로 만드는 방법은 여러 가지가 있으나, 본 논문에서는 별다른 프로그램이 없이도 RDF 트리플을 생성할 수 있는 SQL 쿼리를 이용한다.

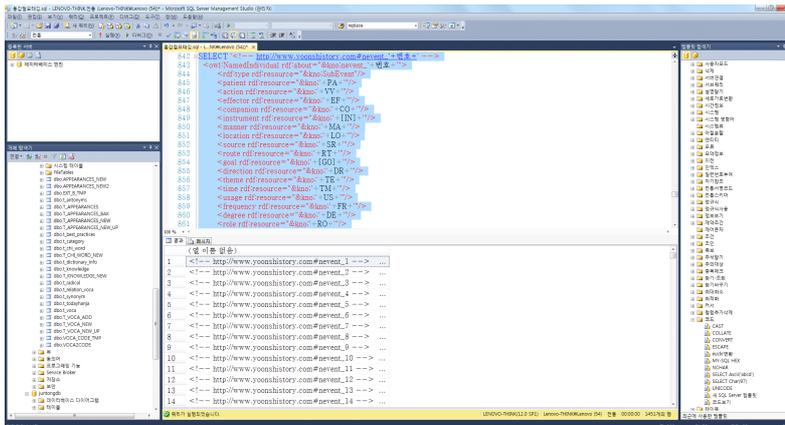


그림 54. SQL 쿼리를 이용한 RDF 트리플의 생성

위에서 나왔던 ‘十一月 中尉韓全誨等 陳兵殿前 請幸鳳翔’를 OWL 문서로 변환하면 다음과 같다.

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#event_450 -->
<owl:NamedIndividual rdf:about=""&kno;event_450">
  <rdf:type rdf:resource=""&kno;Event"/>
  <kno:agent rdf:resource=""&kno;Person;韓全誨"/>
  <kno:action rdf:resource=""&kno;請"/>
  <kno:patient rdf:resource=""&kno;Person;唐昭宗李擘"/>
  <kno:theme rdf:resource=""&kno;event_451"/>
  <kno:time rdf:resource=""&kno;901"/>
</owl:NamedIndividual>
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#event_451 -->
<owl:NamedIndividual rdf:about=""&kno;event_451">
  <rdf:type rdf:resource=""&kno;Event"/>
  <kno:agent rdf:resource=""&kno;Person;唐昭宗李擘"/>
  <kno:action rdf:resource=""&kno;幸"/>
  <kno:time rdf:resource=""&kno;901"/>
  <kno:goal rdf:resource=""&kno;Place;鳳翔"/>
</owl:NamedIndividual>

```

다음은 Protégé로 저장한 문서를 에디터에서 열어 적당한 위치에 위의 문장을 붙여 넣는다. 이때 OWL 문서에서 식별자가 같을 경우에 Protégé에서 자동으로 알아서 병합을 해준다. 이를 OWL 문서에 붙여 넣고 Pertege에서 열면 다음과 같이 나타난다.

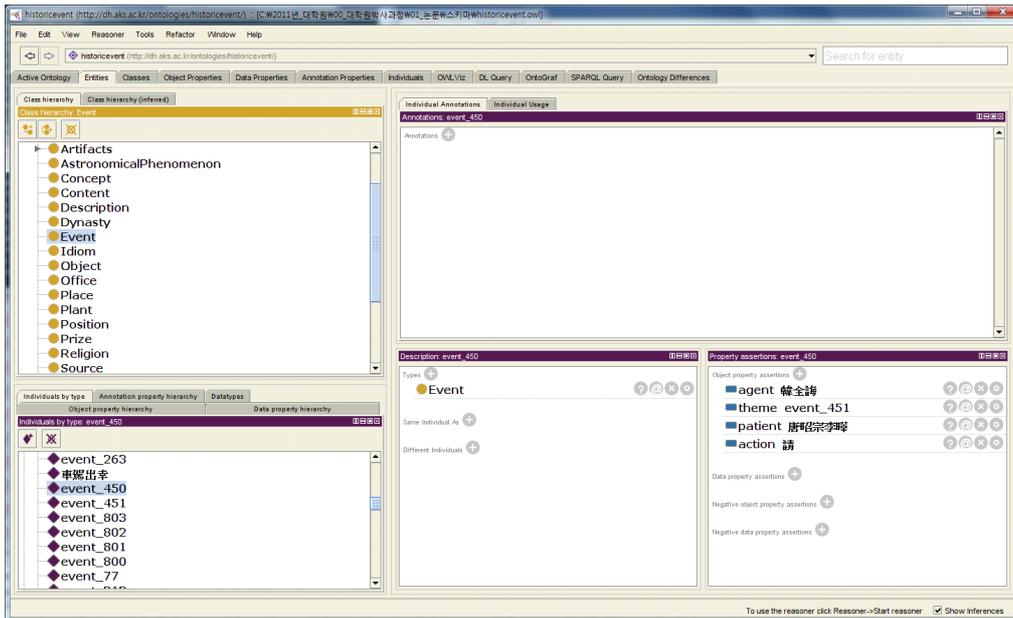


그림 55. Pretege에서 OWL을 불러들인 화면

같은 방법으로 사람 간의 관계나 기타 관계를 RDF 트리플로 변환한다.

2) URL 부여

가) 유일 식별자의 생성

자원을 식별하기 위한 URI는 하나의 개체에 대해서 무조건 하나의 URI를 부여해야 하는 것은 아니다. 하나의 개체에 오직 하나의 URI를 부여한다는 것은 현실적으로 가능하지 않을 뿐더러 그럴 필요도 없다.³⁵⁵⁾ 그러나 하나의 데이터 셋에서는 가끔적이면 같은 자원은 같은 식별자를 부여해 주어야 한다. OWL에서 제공하는 ‘owl:sameAs’는 주로 외부 자원과 연결시키는 것이 그 목적이기 때문이다.

URI는 가능하다면 단순한 형태가 좋다. LOD를 발행하는 목적은 지식을 기계가 이해할 수 있게 하고자 하는 것이지만 궁극적으로 사람이 이해할 수 있게 하는 것도 중요하다. URI 참조 체계가 체계적으로 관리되지 않으면 현실적으로 관리하는 사람들이 힘들기 때문에 일정한 체계를 갖추어야 한다. 어휘를 통해 역사적 지식을 연계 및 통합하기 위해서는 개별 어휘에 대해 고유한 식별자를 발행해야 한다. URI 생성과 관련하여 본 논문에서는 한

355) 오원석, 『Linked Data 4 Principles』, (주)탐쿼드란트코리아, 2015, 10쪽.

문에서 사용된 동사는 한자코드를 그대로 사용하였으며, 다른 명사는 대표명칭을 부여하였다.

그리고 우리말 동사의 경우에는 『표준국어대사전』³⁵⁶⁾의 구분 번호를 사용하였다. 그리고 『通鑑節要』에 나오는 동사는 번역문에 사용된 우리말 동사로 정확하게 연결해 준다. 아래는 『譯註 通鑑節要』에 사용된 다의어 동사이다. 다의어라는 것은 어휘가 한가지 이상의 뜻을 지니고 있는 것을 말한다. 대부분 어렵지 않게 이해할 수 있는 어휘이나 중요한 것은 이들 어휘에는 한가지 뜻만 아니라 여러 가지 뜻을 추가적으로 지니고 있다는 것이다.

가다01, 가두다01, 가하다01, 간하다01, 거두다02, 거하다02, 걸리다01, 결단하다01, 결정하다01, 고치다01, 고하다01, 귀부하다02, 나다01, 내리다01, 내부하다01, 노하다01, 당하다01, 대다01, 대과하다01, 대하다02

이들을 구별해 주지 않으면 정확한 의미 전달이 사실상 어려워진다. 본 논문에서는 『표준국어대사전』의 구분 번호를 URI 참조로 사용하였다. 그 이유는 향후에 지식 관계망을 확대하기 위해서는 역사서에 출현하는 어휘뿐만 아니라 현대 국어 어휘를 모두 포함해야 하기 때문이다. 현대 어휘는 현재 국립국어원에서 서비스하고 있는 『표준국어대사전』³⁵⁷⁾이 가장 많은 어휘를 제공하고 있으며, 세종말뭉치, 고려대학교에서 제공하는 물결21 말뭉치, 울산대학교 말뭉치 등에서 이 구분 번호를 적용하고 있다.

『표준국어대사전』의 구분 번호를 사용해야 하는 또 다른 이유가 있다. 『譯註 通鑑節要』는 물론이고 역사서에는 『표준국어대사전』에 등록되지 않은 수많은 어휘가 나타난다. 주로 한자어에 ‘하다’가 붙은 유형이 대부분인데 역사적 지식을 RDF 트리플로 표현하기 위해서는 이들 어휘까지도 모두 표현해 주어야 한다. 문제는 이런 작업은 어느 개인이 하기에는 쉽지 않다는 점이다. 현실적으로 가능한 방법은 『표준국어대사전』의 구분 번호를 확장하는 방법이다. 『표준국어대사전』에 있는 구분 번호를 유지하면서 추가되는 어휘는 순서대로 구분 번호를 추가하는 것이다. 다행스럽게 전통문화연구회에서는 『조선왕조실록』과 전통문화연구회에서 고전번역서에서 출현한 한자를 모두 취합하여 고유한 식별자를 부착하는 작업을 진행해 오고 있다. 현재까지 조선왕조실록과 전통문화연구회에서 간행한 번역서에서 출현한 한자어휘를 포함하여 모두 80만 개의 어휘에 대해서 식별자를 부여하고 있으며³⁵⁸⁾ 앞으로도 지속적으로 확대해 나갈 예정이다.

356) 사전으로는 『표준국어대사전』 이외에도 세종계획 21의 결과로 만들어진 세종전자사전이 있다.

357) 표준국어대사전, 국립국어원, <http://stdweb2.korean.go.kr/main.jsp>

358) 동양고전번역지원, 전통문화연구회, <http://lu.juntong.or.kr/juntong>

나) 대표명칭의 기술

역사적 사실을 RDF 트리플 형태로 정리하기 위해서는 본문 중에 태깅된 개체명을 대상으로 대표명칭을 부여해야 한다. 본문 속에서 하나의 개체가 서로 다른 이름으로 나타나는 경우가 많기 때문이다. 이들을 정리하지 않으면 온톨로지로 통합할 때 서로 같은 인물에 대한 정보가 통합이 되지 않는다. 동사를 비롯한 일반적인 어휘는 『표준국어대사전』의 구분번호를 이용하여 어느 정도 URI를 생성할 수 있으나, 인명의 경우 식별자를 부여하는 것은 쉽지 않다. 왜냐하면 역사서에 등장하는 인명은 『표준국어대사전』에 등재되지 않은 이름이 많기 때문이다. 한 사람의 인명은 字, 號, 諡號, 廟號, 堂號, 法號, 法名, 初名, 初字, 本名, 別稱 등 다양한 형태로 나타난다. 이들의 성명을 제외하고는 대부분 『표준국어대사전』에 등재되어 있지 않다. 때로는 上, 皇帝, 帝, 主 등과 같이 人名이 아니면서도 사람을 지칭하는 호칭이 있다. 이런 호칭은 人名, 地名, 書名, 官職名을 기본으로 하는 전통적인 고유명사 마크업 방식에서는 태깅의 대상 범위에 포함되지 않았었다.³⁵⁹⁾ 그러나 역사적 지식을 RDF로 표현한다면 이야기가 달라진다. 어떤 형태로 나타나든지 그 행위의 주체는 중요한 의미가 있기 때문에 개체명 식별 단계에서 반드시 마크업의 범위에 포함시켜야 이후 단계에서 부담을 줄일 수 있다. 또한 代表名稱은 人名, 地名, 書名이나 기타 모든 클래스에 모두 적용되는 개념이다. 사실 代表名稱은 개체의 고유성을 확보뿐만 아니라 작업의 편의성 또한 고려한 개념이다.

우선 봉건 시대의 황제는 諡號 앞에 廟號가 있었다. 漢代 이후 매 왕조의 첫째 皇帝는 일반적으로 太祖·高祖 혹은 世祖라고 불렀고, 이후의 황제들은 太宗·世宗 등으로 불렀다. 漢代에는 모든 皇帝가 廟號를 가진 것이 아니라 ‘有功, 有德’해야만 비로소 祖나 宗이라고 불리었다. 南北朝 시대에는 宗이라고 부르는 것이 흔해졌고, 唐대에 이르러서는 宗이라고 칭하지 않는 황제가 없게 되었다. 당대 이전에는 죽은 임금에 대해서 간단히 諡號를 불렀고 廟號를 부르지 않았으나, 唐代 이후에는 諡號가 더 길어짐에 따라 부르기가 불편하였으므로 唐玄宗, 宋太祖와 같이 廟號로 불렀다. 年號는 봉건황제의 紀年の 명호이다. 年號는 漢武帝 때 처음 생겼는데 漢武帝가 즉위한 해를 建元 원년(B.C. 140)이라고 불렀으며 다음해는 建元 2년이라고 불렀다. 새로운 임금이 즉위하면 반드시 年號를 바뀌었는데 이것을 改元이라 한다. 같은 황제가 재위 중에도 改元할 수 있었다. 예를들면 漢武帝는 재위 중에 建元, 元光, 元朔, 元狩, 元鼎, 元封, 太初, 天漢, 太始, 征和, 後元 등 여러 번 연호를 바꾸었다. 明·清 황제들은 기본적으로 改元을 하지 않았으므로 年號로 황제를 부를 수가 있었다. 예를들면 明世宗은 嘉靖皇帝, 清高宗은 乾隆皇帝로 불렀다.³⁶⁰⁾

359) 조선왕조실록, 승정원일기, 향토문화전자대전, 기타 중국의 표점 지침에서도 일체 범위에 포함시키지 않았다.

이런 점을 고려하여 唐代 이전의 황제는 ‘王朝+諡號+姓名’로 唐代부터는 ‘王朝+廟號+姓名’의 형태로 대표명칭으로 정의한다.

위에서 언급한 것이 인명의 다양한 유형에 대한 해결 방안이라면 다음은 同名異人의 처리에 대한 방안이다. LOD에서 모든 자원은 URI를 활용하여 표현하며 URI는 웹상에서 유일하게 식별되어야 하기 때문에 반드시 同名異人을 구분해 주어야 한다는 점이다. 이를 해결하기 위해서 보통 일련번호 등의 인위적인 ID를 부여하기도 한다. 그런데 위와 같은 방법은 인스턴스의 유일성을 확보할 수 있으나 그 의미성을 잃어버리게 된다. 이를 위해서 본 논문에서는 위에서 언급한 대표명칭을 사용하며 만약 중복이 발생하면 일련번호로 구분해 주었다. 예를 들어 ‘後周世宗郭榮’은 ‘http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#後周世宗郭榮’과 같이 유일하게 참조된다.

```
<owl:NamedIndividual rdf:about="&knoPerson;後周世宗郭榮">
  <rdf:type rdf:resource="&kno;Person"/>
  <kno:aliasName rdf:datatype="&xsd:string">榮</kno:aliasName>
  <kno:name rdf:datatype="&xsd:string">郭榮</kno:name>
</owl:NamedIndividual>
```

3) RDF 트리플 생성

가) event의 구성

역사적 사건은 시간 간격이 긴 것과 짧은 것으로 구분할 수 있다. 그렇기 때문에 『通鑑節要』에 기록된 다양한 역사적 사건을 한 가지 레벨의 기술문으로 표현하기에는 어려움이 있다. 큰 사건 안에 다수의 작은 사건이 존재하고 작은 사건은 주로 행위에 의해 구체화된다. 본 연구에서는 하나의 큰 사건과 큰 사건을 구성하는 작은 사건, 그리고 작은 사건을 구성하는 행위를 기술할 수 있도록 설계하였다. 큰 사건과 작은 사건, 그리고 구체적인 행위는 모두 Event 클래스로 분류하고, 이들 사건 간의 관계는 subEvent와 superEvent로 서로 연결한다. 사건을 구성하는 구체적인 행위를 표현하는 프로퍼티는 Object Property와 Datatype Property'로 모두 구성한다. 그 이유는 event와 연결되는 목적어는 클래스로 정의된 것뿐만 아니라 리터럴로 표현해야 하는 것도 있기 때문이다. 楚漢戰爭으로 예를 들어 보자. 楚漢 戰爭은 5년 동안의 전쟁이고 그 중에서 하나를 구성하는 것이 彭城戰鬪이다. 『通鑑節要』의 B.C.205(병신)년의 기사에 따르면 彭城戰鬪는 아래와 같이 사건이 전개된다.

360) 王力(지)/李鴻鎮(역), 위의 책(1994), 107쪽.

彭越 將兵歸漢 漢 遂入彭城 收其貨寶美人 日置酒高會 項王 聞之 自以精兵三萬人 至彭城 大破漢軍 於睢水 漢軍 爲楚所擠 卒十餘萬人 皆入睢水 睢水爲之不流

彭越이 병력을 인솔하고 漢나라에 귀의하자, 漢나라가 마침내 彭城에 들어가서 그 보화와 미인을 거두고 날마다 술자리를 베풀고 크게 모여 잔치하였다. 項王이 이 말을 듣고 정예병 3만 명을 거느리고 彭城에 이르러 漢軍을 睢水에서 대파하였다.

漢나라 군사들은 楚나라 군사들에게 밀려서 병졸 10여만 명이 모두 睢水로 들어가니, 睢水가 이 때문에 흐르지 못하였다.

이 중에서 첫 번째 문장을 RDF 그래프로 표현하면 다음과 같다.

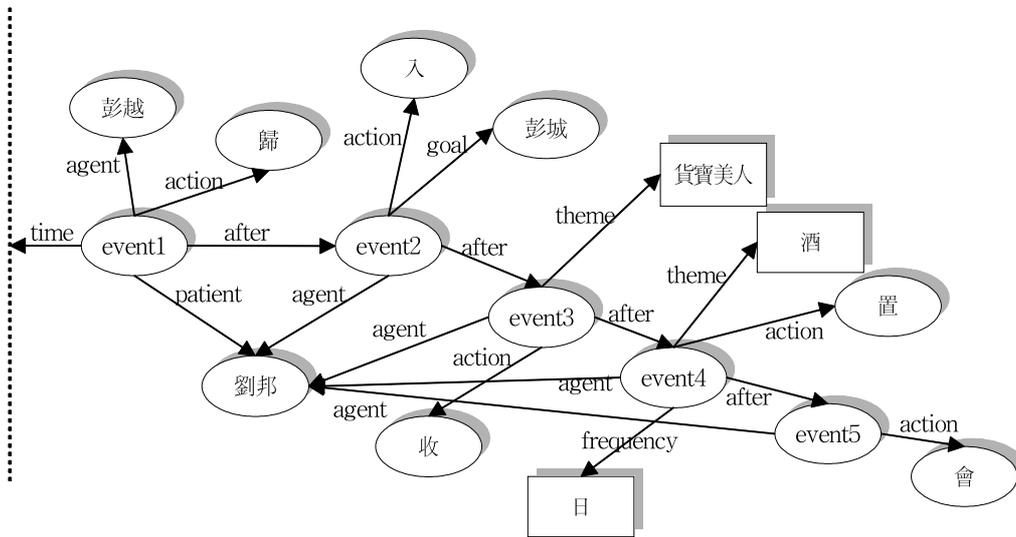


그림 56. 項羽가 劉邦을 睢水에서 대파한 사건의 RDF 그래프 표현

우선 그래프는 위에 문장에서 나오는 동사를 기준으로 모든 행위가 하나의 event를 나타낸다. 개별 event는 각각 선후 관계를 after 프로퍼티를 이용하여 표현한다. 구체적인 행위는 Action 클래스의 인스턴스가 되며, event와 Action의 인스턴스는 action 프로퍼티로 연결이 된다.

(1) 다항 관계의 표현

『通鑑節要』는 編年體로 기술한 역사서이다. 원칙적으로 역사적 사실을 발생한 연도에 따라 기술했기 때문에 문장 내에 구체적인 시간을 기록하지 않았어도 대부분의 행위는 발생한 연도를 파악할 수 있다. RDF 트리플로 표현하고자 하는 사건은 본문에 구체적인 연도가 표현되어 있지 않더라도 관계없이 해당 연도에 발생한 것을 확인할 수 있으면 모두 해

당 연도를 시간 정보와 연결해 준다. 아래 문장은 南康王 蕭寶融이 즉위한 시간과 장소, 국적, 봉작호 관계를 표현한 것이다

- 三月에 齊南康王寶融이 卽位於江陵하다
- ☞ 3월에 齊나라 南康王 蕭寶融이 江陵에서 즉위하였다.

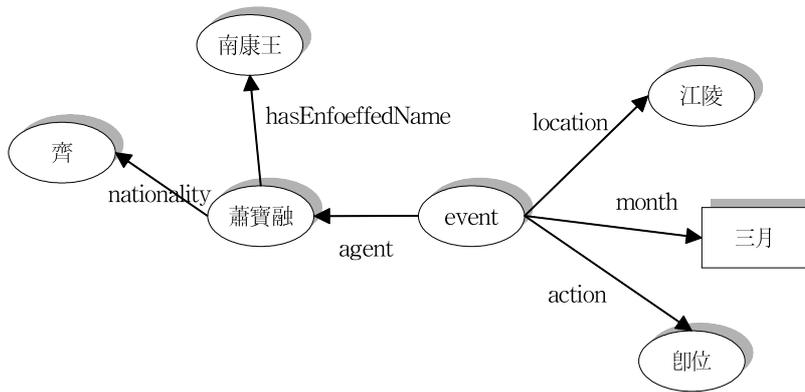


그림 57. event와 의미역을 활용한 다항 관계의 표현

(2) 주어와 목적어

한문 문장이나 이를 번역한 문장에서는 주어나 목적어가 생략되더라도 문장을 이해하는데 큰 지장이 없는 경우가 많다. 그렇지만 RDF 트리플에서는 모든 관계를 주어-술어-목적어로 표현해 주기 때문에 특별한 경우가 아니고는 모든 주어나 목적어를 표현해 주어야 한다. 그래야 주어와 목적어를 통해서 RDF 문장을 통합하거나 확장해 나갈 수 있기 때문이다.

주어는 행동주와 영향주로 표현할 수 있다. 주어의 위치에 올 수 있는 것은 사람뿐만 아니라 사물이 될 수도 있다. 때로는 단체, 국가 등이 주어로 오는 경우가 있다. 아래의 두 개 문장은 주어가 漢과 楚일 수도 있고, 漢軍과 楚軍으로 볼 수도 있으며, 군대를 이끄는 장수로 볼 수도 있다. 이런 경우에는 정해진 규칙은 없지만 가급적 사람으로 표현하고자 하였다. 그럴 경우 아래의 첫 번째 문장에서 실질적 주어는 ‘韓信’이고 두 번째 문장에서 나오는 군대는 ‘曹咎’이다.

- 漢 數挑戰

☞ 漢나라가[韓信] - 싸움을 걸다 - (楚나라에)[曹咎]

■ 楚軍 不出

☞ 楚나라 군대가[曹咎] - 나오지 않다[응하지 않다] - (城)[城]

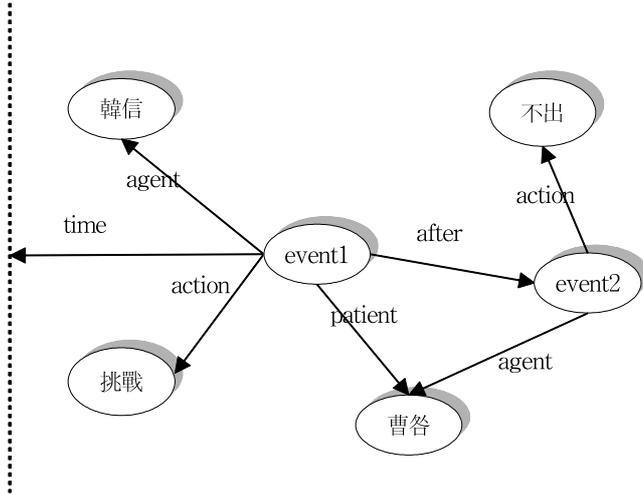


그림 58. 실제 주어로 표현한 그래프

복수의 주어가 나타나는 경우도 흔하다. 아래 문장은 주어가 李茂貞과 朱全忠이 된다. 이 때 李茂貞을 행동주로 놓고 朱全忠을 동반주로 볼 여지도 있으나 이 문장에서는 서로 동등한 자격을 갖추었으므로 모두 行動主로 보았다.

■ 李茂貞 與朱全忠 皆欲發兵 迎天子

☞ 李茂貞과 朱全忠이 모두 군대를 동원하여 천자를 맞이하려 하자,

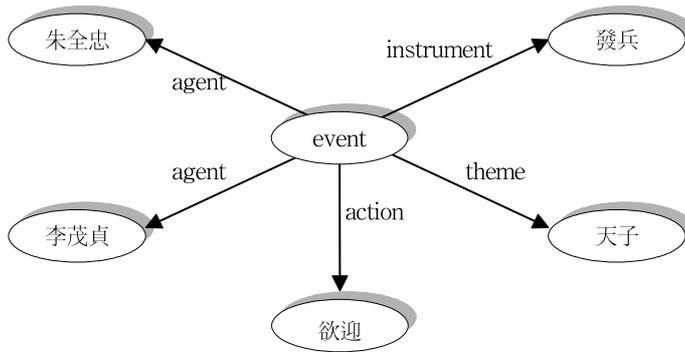


그림 59. 복수의 주어 표현

아래 문장은 주어가 霍顯과 霍禹, 霍山, 霍雲 등 4명이나 된다.

- 霍顯及禹, 山, 雲 自見日侵削 數相對啼泣自怨 謀廢天子 事發覺
- ☞ 霍顯과 霍禹, 霍山, 霍雲이 날마다 侵削당하는 것을 스스로 보고는 자주 서로 마주 보고 울며 스스로 원망하여 天子를 폐위할 것을 도모하다가 일이 발각되었다.

역사서에서 주어가 나타나지 않는 경우는 흔히 나타난다. 아래 문장에서는 행동주인 ‘唐僖宗 李儂’이 생략된 형태이며, 피동주는 ‘田令孜’이다.

- 削田令孜官爵 長流端州
- ☞ 田令孜의 관작을 삭탈하여 멀리 端州로 유배보냈다.

다음은 목적어의 생략에 대해서 살펴 본다. 아래 문장에서 ‘籍’는 등록하다, 기재하다라는 의미로 사용되었으며 목적어를 표현하면 ‘장부’가 될 것이다.

- 今既籍爲官物 非有詔書 不可得也
- ☞ 이미 장부에 기재하여 관청의 물건이 되었으니 황제의 조서가 있지 않으면 가져갈 수 없다

아래 문장에서는 목적어가 분명하게 드러나 있지 않다. 그러나 번역문을 참조하여 적절한 목적어를 보충해 넣는다.

- 巢入城 勞問而已 閭里晏然
- ☞ 黃巢가 도성에 들어가 단지 위로하기만 하니 마을이 편안하였다.

아래 문장은 목적어가 ‘第五可範’과 ‘數百人’이 된다.

■ 是日 全忠 以兵驅第五可範已下數百人於內侍省 盡殺之

☞ 이날 朱全忠이 군대를 이끌고 환관인 第五可範 이하 수백 명을 內侍省에서 몰아내어 모두 죽이니,

아래는 의미역 중 대상(theme)으로 분류한 어휘이다.

■ 1. 朱全忠以梁兵[agent] 李克用以晉兵

■ 2. 梁兵[agent] 潰圍出 擒王彥章 斬之

위의 두 개 문장에서 행동주인 梁兵이다. 梁兵은 다른 왕조의 병사와 구분하기 위해 後梁兵으로 하였다. 그런데 위 문장에서 두 가지 문제를 제기할 수 있다. 첫 번째는 두 개의 집단을 같다고 볼 수 있느냐는 것과, 두 번째는 그 범위가 불확실한 집단을 인스턴스로 등록할 수 있느냐 하는 것이다. 1번에서 兵은 朱全忠이 이끈 後梁의 병사이고 2번은 朱全忠의 아들인 朱友貞이 이끈 병사이다. 이 두 개의 집단은 그 구성원으로 본다면 서로 같다고 할 수 없다. 그러나 이런 집단을 서로 다른 것으로 보면 모든 집단을 다 다른 것으로 파악해야 한다. 이것은 매우 비효율적이다. 비록 後梁兵이 구체적인 명칭이 없는 일반적인 집단이라 하더라도 개념적으로 보아 일정한 집단으로 이해해야 한다.

다음으로, 後梁兵은 분명하게 행동주의 역할을 하지만 그 범위가 모호한데 이를 별도의 URI로 식별할 것인가는 관리의 문제로 접근해야 할 것으로 보인다. 이 문제는 後梁의 병사와 같은 복합명사를 개체로서 인정하면 그 수가 무한정 늘어나기 때문에 독립적인 개체로 인정하기 어려운 것으로 생각할 수 있다. 그러나 後梁兵과 같은 유형은 일정한 실체가 있기 때문에 그들을 하나의 개체로 인정하는 것도 나쁘지는 않다고 생각한다. 본 연구에서도 이들을 Group으로 분류하였다.

다음은 유사한 용어를 어떻게 정리해야 하는가에 대한 문제이다. 아래 문장에서 보면 ‘梁人’과 ‘梁兵’은 의미적으로 동일하게 ‘後梁의 군사’를 의미한다. 아마 저자도 특별한 의미를 부여하지 않고 그때그때 적절한 용어를 사용했을 것이다.

■ 梁人 擊敬瑄 斷其馬甲

■ 梁兵 潰圍出 擒王彥章 斬之

이들을 모두 인스턴스로 등록하는 것은 그 숫자가 너무 많아 지기때문에 이들 어휘는 의미별로 그 유형을 모두 분류하고 그 의미에 적합한 어휘를 사용하는 것이 가장 바람직하다. 그러나 이들을 의미별로 통일시키는 것은 너무나 많은 부담이 된다. 일정한 기준에 의해 가급적 통일된 어휘를 사용하되 그 기준은 아무래도 국역본이 되어야 할 것이다.

다음은 주어의 일부를 구성하고 있는 사람의 문제이다. 아래 문장에서는 주어는 ‘宋審虔

四將’인데 四將을 어떻게 표현할 것인가이다. 주어를 宋審虔 하나만을 표현하기는 실제 의미와 달라지므로 문제가 된다. 이를 표현할 수 있는 적절한 방법을 찾을 수 없어 임시로 이런 유형의 주어를 표현할 수 있는 ectSubject를 두었으며 이를 통해서 표현한다.

■ 唐主又與宋審虔等四將 議復向河陽

☞ 唐主가 또 宋審虔 등 네 장수와 함께 다시 河陽으로 향할 것을 의논하였는데,

(3) 상태나 현상의 표현

자동사는 목적어가 필요 없는 동사를 말한다. RDF 트리플은 반드시 주어와 술어, 목적어가 존재해야 한다. 그렇기 때문에 목적어가 없는 문장은 공백 노드를 사용하여 트리플을 구성해야 한다. 아래 문장에서는 日食이라는 어휘 자체에 이미 일식이 나타남을 표현하고 있다.

■ 十二月朔 日食하고

☞ 12월 초하루에 日食이 있었고



그림 60. 목적어가 없이 주어와 술어만 존재하는 경우

이런 유형을 표현하기 위해 ‘event’와 ‘日食’의 관계를 ‘effector’로 표현하며, ‘나타나다’는 ‘action’이라는 프로퍼티를 사용하여 표현한다.

■ 日有食之

☞ 일식이 있다

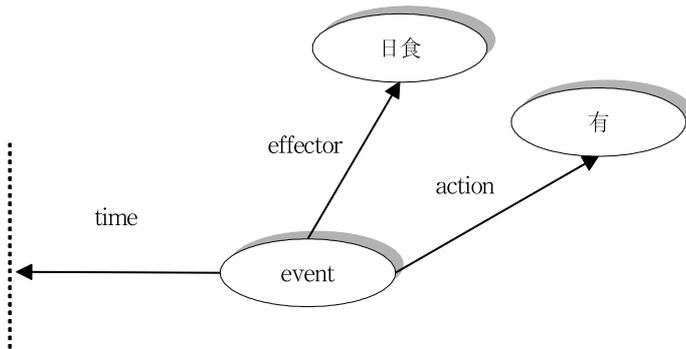


그림 61. 영향주에 의한 상태의 표현

때로는 어떤 행위의 결과로 발생한 상태를 표현해야 할 경우가 있다. 아래 문장은 군세가 더욱 강성해졌다는 event2의 결과로 어떤 상태에 놓인 것이다.

■ 朱全忠 大破秦宗權 斬之 全忠 既克蔡州 兵勢益盛

☞ 朱全忠이 秦宗權을 크게 격파하여 그의 목을 베었다. 朱全忠이 蔡州를 점령하니, 군세가 더욱 강성하였다.

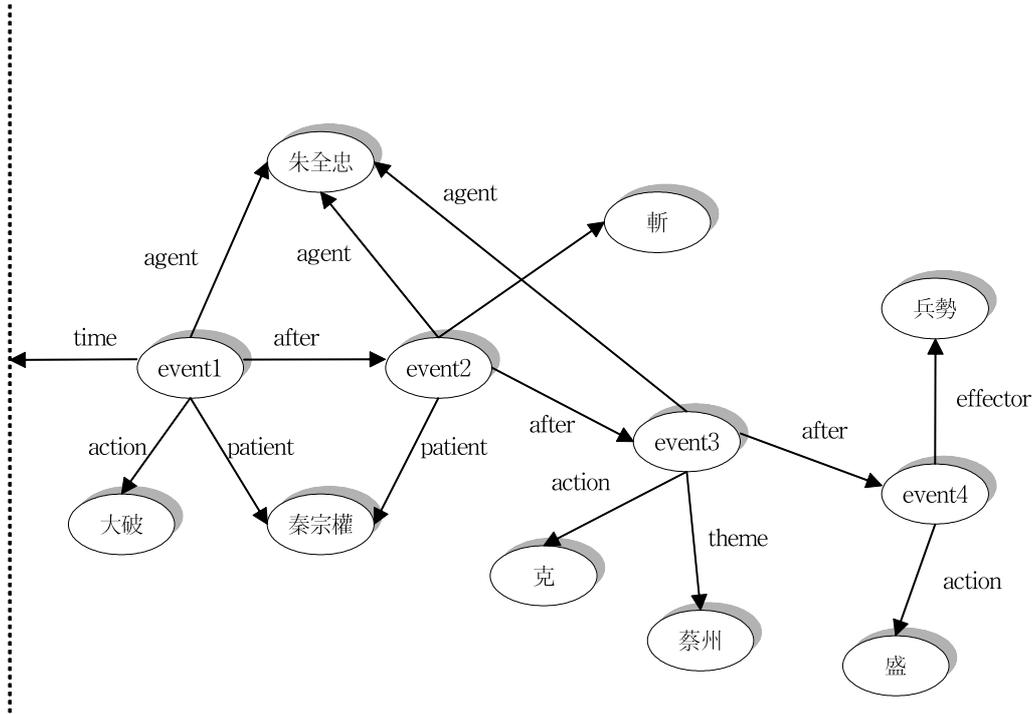


그림 62. 朱全忠이 蔡州를 점령하니, 군세가 더욱 강성해짐을 표현하는 그래프

아래 첫 번째 문장은 단기간에 발생한 사건이며, 두 번째 문장은 오랜 기간 동안 지속된 상태를 나타낸다. during은 사건이나 상태의 지속을 나타내는 값과 연결시키는 프로퍼티이다.

- 三月 齊南康王寶融 卽位於江陵(3월 중 어느 한 날)
 - ☞ 3월에 齊나라 南康王 蕭寶融이 江陵에서 즉위하였다.
- 隋文 在位二十三年
 - ☞ 隋나라 文帝가 23년 동안 재위하였으니

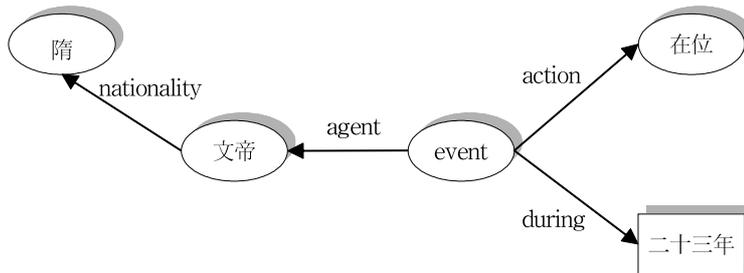


그림 63. 隋나라 文帝가 23년 동안 재위한 상태를 표현하는 그래프

(4) 동사가 연속되는 경우

아래의 문장의 ‘습격하다’와 ‘격파하다’는 두 개의 event이면서 사실상 거의 하나의 행위로 볼 수도 있다. 이러한 관계는 연속 동작임을 나타내는 ‘continuousAction’ 프로퍼티를 이용하여 표현하였다. ‘continuousAction’은 event의 선후 관계에 비해 그 밀접도가 높음을 표현한다. Person의 인스턴스인 ‘冒頓單于’과 Tribe의 인스턴스인 ‘匈奴’의 관계는 ‘hasRelate’를 이용하여 연결하였다. 역사 자료에서 출현하는 수많은 관계를 모두 프로퍼티로 표현하기는 어렵기 때문에 중요한 관계를 제외하고는 모두 ‘hasRelate’로 표현하였다.

■ 匈奴冒頓單于襲破之 餘衆 保鮮卑山

☞ 匈奴의 冒頓單于가 습격하여 격파하니, 남은 무리가 鮮卑山을 확보하였다.

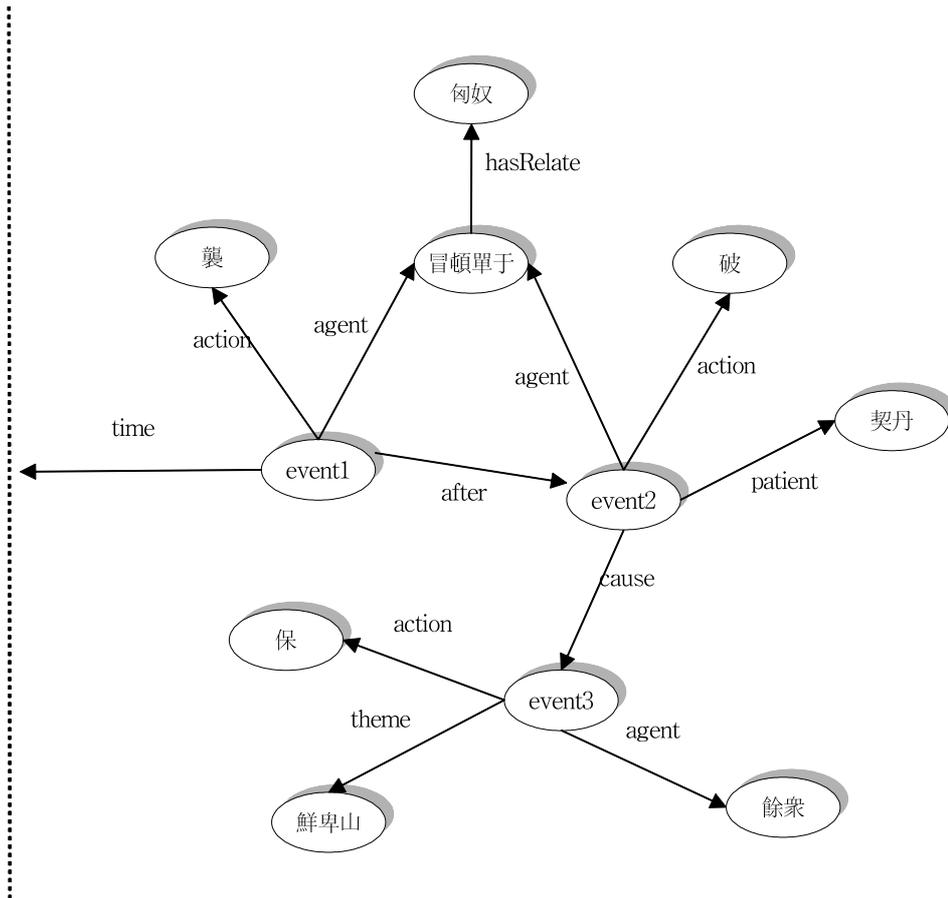


그림 64. 匈奴의 冒頓單于가 거란을 습격한 사건을 표시하는 그래프

(5) 사동, 사역

역사서에는 누군가 다른 사람에게 강요하는 행위가 적지 않게 나타난다. ‘하게 하다, 시키다, 하여금 하게 하다’ 등이 그러한 예인데 이러한 어휘는 정도의 차이가 있으나 모두 ‘시켰다’는 의미로 표현이 가능하다.

일반적으로 어떤 대상에게 어떤 행위를 하게 하는 사동문은 직접 사역과 간접 사역으로 구분할 수 있다. 직접 사역은 ‘행동주’가 사역을 나타내는 동사에 의해 직접적으로 ‘피동주’에게 어떤 행위를 하거나 피동주가 어떠한 상태에 이르게 하는 것인 데 비해, 간접 사역은 ‘사동주’가 ‘피동주’에게 어떤 행위를 하게 하는 것이다.³⁶¹⁾ 아래 문장을 살펴 보면 첫 번째

361) 사동 문장의 주어(를) 사동주라고 하고 사동주의 어떤 행위에 의해 다른 행위를 하게 되는 주체를 피사동주라고 한다.

문장은 'A가 B를 ~하여 C로 하다' 형식으로 되어 있으며 두 번째 문장은 'A가 B를 시켜 C를 어떻게 하게 하다'라는 형식으로 되어 있다. 아래 첫 번째 문장과 두 번째 문장은 '삼다'라는 행위의 주체는 '周市'가 되나, 두 번째 문장은 '시해하다'의 행위의 주체는 '朱全忠'이 아니라 '蔣玄暉'가 된다.

■ 周市立魏公子咎 爲魏王

☞ 周市[agent]가 魏나라 公子 咎[patient]를 세워[action] 魏王[goal]으로 삼았다[action].

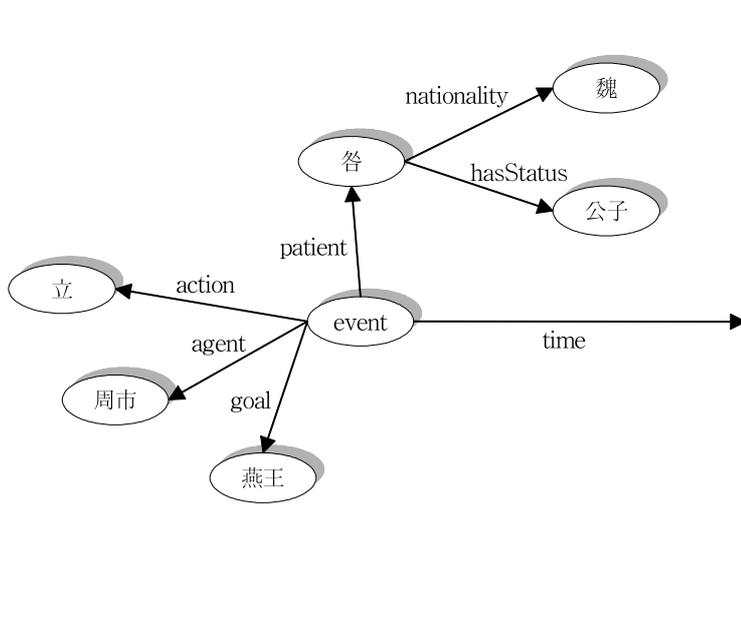


그림 65. 周市가 魏나라 公子 咎를 세워 魏王으로 삼았다 사건을 표현하는 그래프

다음의 문장도 동일한 구조이다.

■ 全忠 使蔣玄暉 全忠以爲樞密使 弑之 立輝王 爲皇太子

☞ 朱全忠[agent]이 蔣玄暉[patient]로 하여금 황제를 시해하게[action] 하고[action] 輝王[patient]을 皇太子[goal]로 세웠다[action].

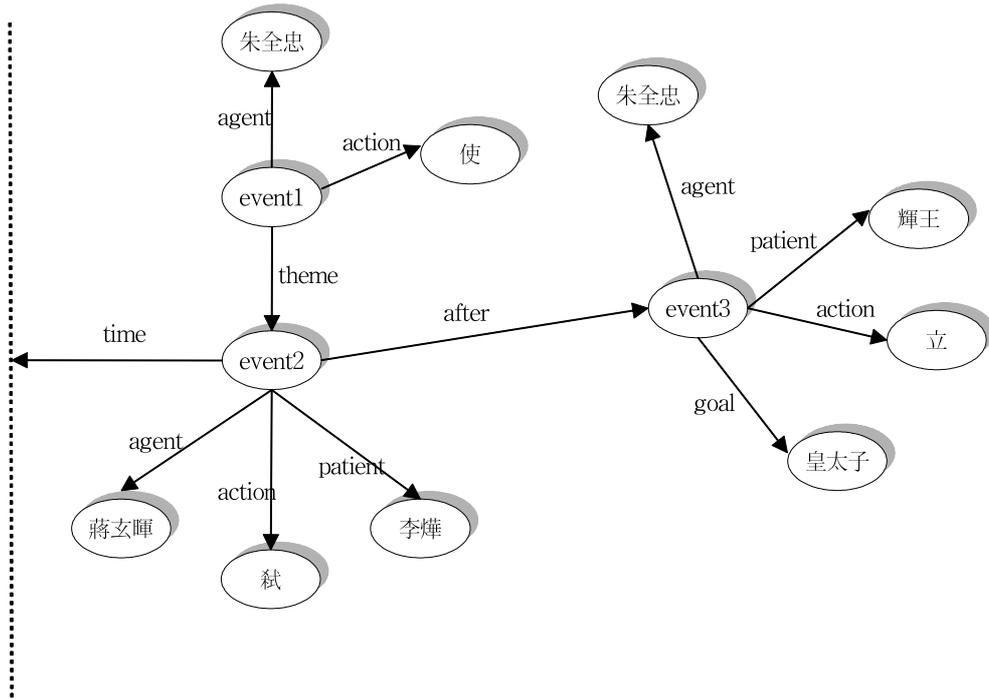


그림 66. 朱全忠이 황제를 시해하고 輝王을 皇太子로 세운 사건을 표현하는 그래프

아래 문장에서는 행동주는 ‘上’이 되고 피동주는 ‘杜讓能’이고, 杜讓能에게 전달하도록 한 대상 업무는 ‘李茂貞을 토벌하는 일’이 된다. 꽤 복잡한 구조이지만 역시 표현하는 데 큰 문제는 없다.

- 李茂貞이 恃功 驕橫 上怒 欲討之 命杜讓能 專掌其事
- ☞ 李茂貞이 공로를 믿고 교만하고 제멋대로 행동하자, 上이 노여워하여 그를 토벌하고자 하였다. 그리하여 杜讓能에게 명해서 이 일을 전담하게 하였다.

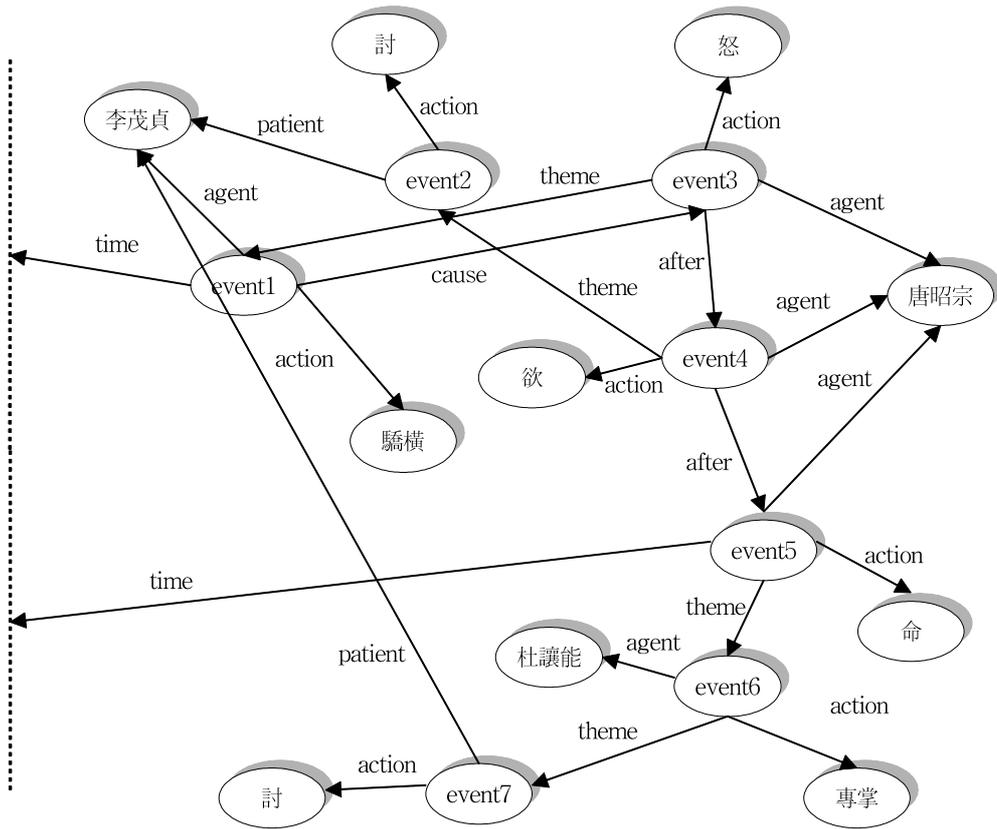


그림 67. 杜讓能에게 李茂貞을 토벌하는 일을 전담하도록 한 내용의 그래프 표현

(6) 행위에 참여하는 참여자

어떤 행위에 참여하는 것을 同伴主(companion)라고 한다. 동반주는 주어의 행위에 참여하거나 공동으로 한다. 앞에서 설명한 것처럼 ‘와/과’로 연결되는 경우 동반주로 식별할 수 있다. 아래 내용은 唐 玄宗 李隆基가 上林苑의 北門을 통해서 달아났다는 것으로, 이때 ‘上林苑의 北門’은 통과한 경로이기 때문에 route가 되며, 행위의 동반주는 ‘王貴妃, 韋淑妃, 太子 및 諸王’ 등이 된다. 이때 동반주의 숫자는 제한이 없다.

■ 上 乃與王貴妃 韋淑妃 太子 諸王 自苑北門出

☞ 上이 마침내 王貴妃, 韋淑妃, 太子 및 諸王과 함께 上林苑의 북문으로 탈출하였다

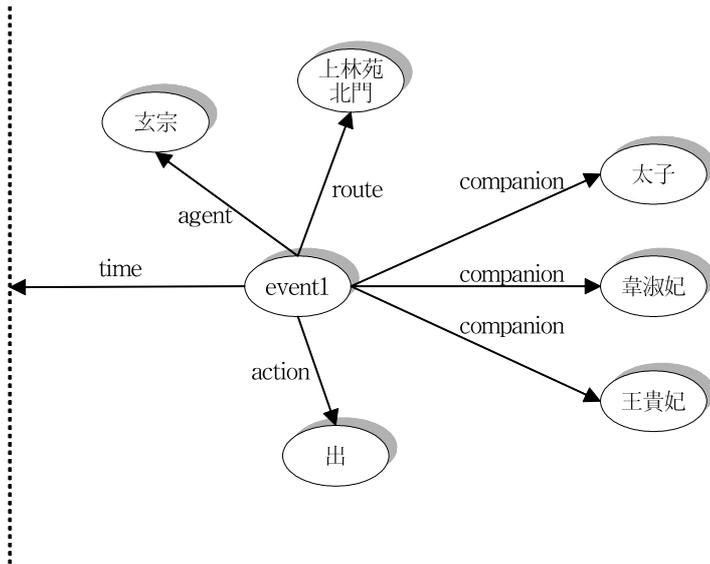


그림 68. 동반주의 표현

(7) 공간, 발생 장소

사건은 반드시 공간에서 발생하기 마련이다. 행위는 행동주와 공간을 직접 연결해 주기도 하나 본 논문에서는 의미역을 이용하여 event가 발생한 시간, 인과 관계 등을 표현하였다. 발생한 장소는 location으로 표현한다.

■ 濮州人王仙芝 始聚眾數千 起於長垣

☞ 濮州 사람 王仙芝가 처음으로 수천 명의 무리를 모아 長垣에서 일어났다.

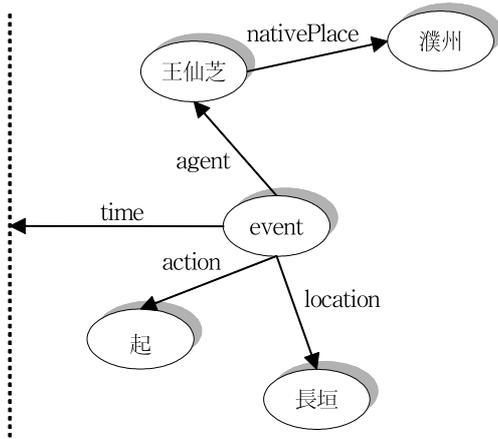


그림 69. 사건이 발생한 장소의 표현

(8) 인과 관계의 표현

역사서에서 인과 관계는 중요한 역할을 한다. 그러나 본문 중에서 인과 관계를 파악하는 것은 쉽지 않다. 내용을 정확하게 읽고 인과 관계를 찾아서 표현해 주어야 한다. 아래 문장은 ‘蘇秦이 趙나라를 떠나다’와 ‘從約이 모두 와해되었다’라는 두 개의 문장으로 구성되어 있으며 이 두 개의 문장은 시간의 선후 관계인 동시에 인과 관계이다. 인과 관계는 ‘cause’를 이용하여 표현하며, 蘇秦은 행동주(agent)이며, 조나라는 蘇秦이 떠난 기점(source)이 된다. ‘從約’은 무생물이면서 주어의 역할을 하기 때문에 ‘effector’로 표현한다.

■ 蘇秦이 去趙하니 而從約이 皆解러라

☞ 蘇秦이 趙나라를 떠나니, 從約이 모두 와해되었다.

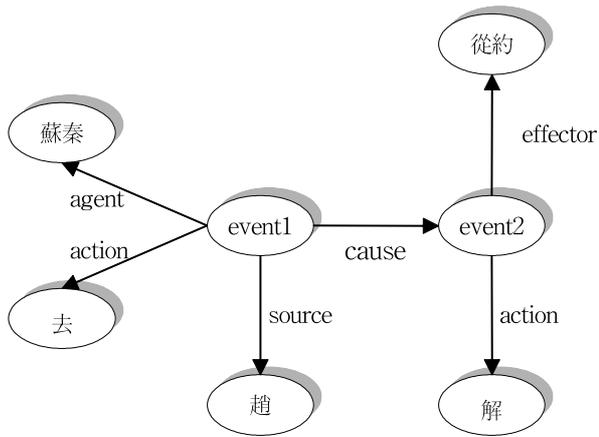


그림 70. 蘇秦이 趙나라를 떠나니 從約이 모두 와해되다의 그래프 표현

아래 문장에서 高明曜와 軍士의 목을 벤 이유는 高明曜가 적의 기생을 데려 오고, 尙可孤의 軍士가 멋대로 적의 말을 취했기 때문이다. event1의 원인은 event2와 event3이 된다. 이들을 cause로 연결해 주었다.

- 晟의 大將高明曜 取賊妓 尙可孤 軍士 擅取賊馬 晟 皆斬之 軍中股慄 公私安堵 秋毫無犯
- ☞ 李晟의 大將인 高明曜가 적의 기생을 데려 오고 尙可孤의 軍士가 멋대로 적의 말을 갖자 李晟이 모두 목을 베니, 軍中이 두려워하여 다리를 떨고 公私가 안도하여 추호도 범함이 없었다.

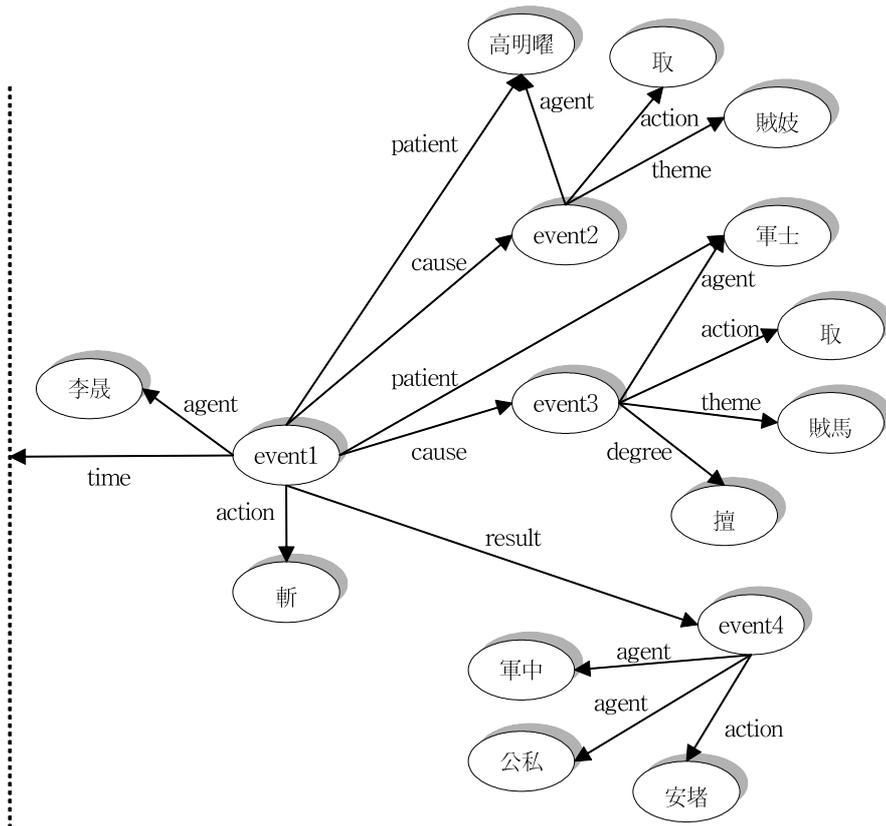


그림 71. 高明曜와 軍士의 목을 벤 사건의 인과 관계 표현

개체의 어떤 행위로 인한 반작용으로 어떤 행위가 발생한다. 아래 문장에서 보면 ‘韓信이 齊王을 추격하다’와 ‘項羽가 龍且를 시켜서 齊나라를 구원하게 하다’ 사이에 인과 관계가 성립한다. ‘추격하다’와 ‘구원하다’는 프로퍼티에도 선후 관계가 발생한다. ‘韓信이 齊王을 추격한 행위’와 ‘項羽가 龍且를 시켜 제왕을 구원하게 한 행위’ 간에 약간의 간극이 있겠지만 인과 관계 뿐만 아니라 시간적으로 선후 관계도 존재한다. 이러한 event 간 인과 관계를 ‘cause’로, event의 선후 관계를 ‘after’로 표현하였다.

■ 韓信 已定臨淄 遂東追齊王 項王 使龍且 將兵救齊

☞ 韓信이 이미 臨淄를 평정하고 마침내 동쪽으로 齊王을 추격하니, 項王이 龍且로 하여금 병력을 거느리고 가서 齊나라를 구원하게 하였다.³⁶²⁾

362) 韓信이 齊나라의 수도 臨淄를 평정하고 도망하는 齊王 田廣을 추적하니 項羽가 龍且로 하여금 田廣을 구원하게 하였다. 龍且는 韓信을 겁쟁이라고 여겼는데 이 싸움에서 龍且가 패하면서 楚漢

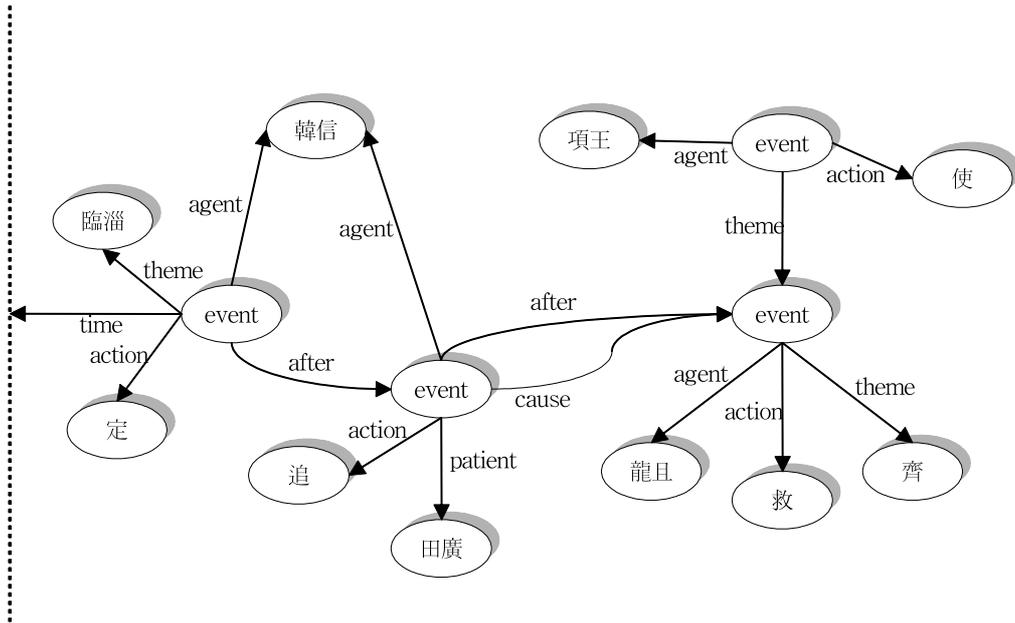


그림 72. 韓信의 event와 龍且의 event의 인과 관계의 표현

(9) 이동이 발생하는 경우

이동을 나타내는 동사에는 주로 ‘기점(source), 도착지(goal), 경로(route), 방향(direction), 대상(Object), 행위자(Agent), 시간(time)’이 나타난다. 아래 문장은 도착지(goal)가 나타나 있다. 이때 문장에서는 직접 언급이 되지 않았지만 행위의 주체는 황제가 된다. 이러한 행동주에 대해서도 파악할 수 있는 경우에 정보를 넣어 주었다.

■ 削田令攷官爵 長流端州

☞ 田令攷의 관작을 삭탈하여 멀리 端州로 유배보냈다.

전쟁은 漢나라로 저울추가 기울게 된다.

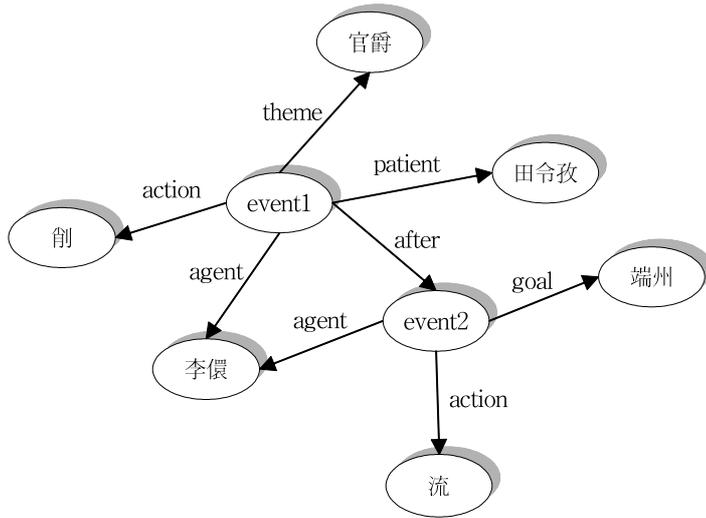


그림 73. 田令孜의 관작을 삭탈하여 멀리 端州로 유배보냈다는 문장의 그래프 표현

아래 문장에서는 ‘도착지(goal), 대상(Object), 행위자(Agent), 시간(time)’이 나타나 있다.

■ 春二月 朱全忠 遣使如契丹

☞ 봄 2월에 朱全忠이 契丹에 使者를 보내었다.

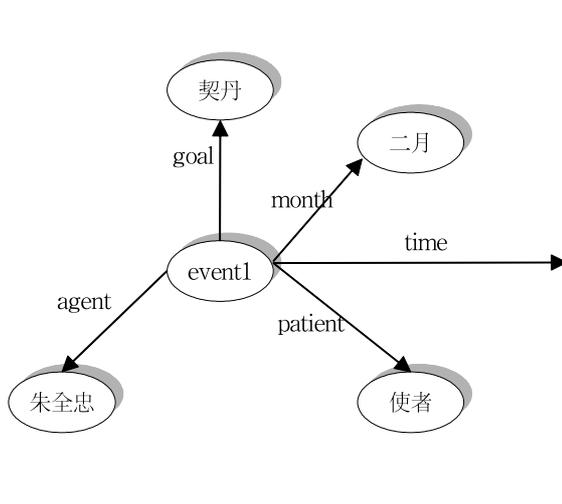


그림 74. 봄 2월에 朱全忠이 契丹에 使者를 보내었다의 그래프 표현

(10) 도구, 방법

사건을 기술할 때 ‘도구나 방법’이 나타나는 경우도 많다. 아래 문장에서는 王仙芝의 수급을 파발마로 전달한 것을 나타냈다.

- 招討使曾元裕 大破王仙芝於黃梅 殺五萬餘人 追斬仙芝 傳首
- ☞ 招討使 曾元裕가 黃梅에서 王仙芝를 크게 격파하여 5만여 명을 죽이고, 王仙芝를 추격하여 목을 베어 首級을 조정에 파발마[instrument]로 전달하였다.

이를 그래프로 표현하면 다음과 같다.

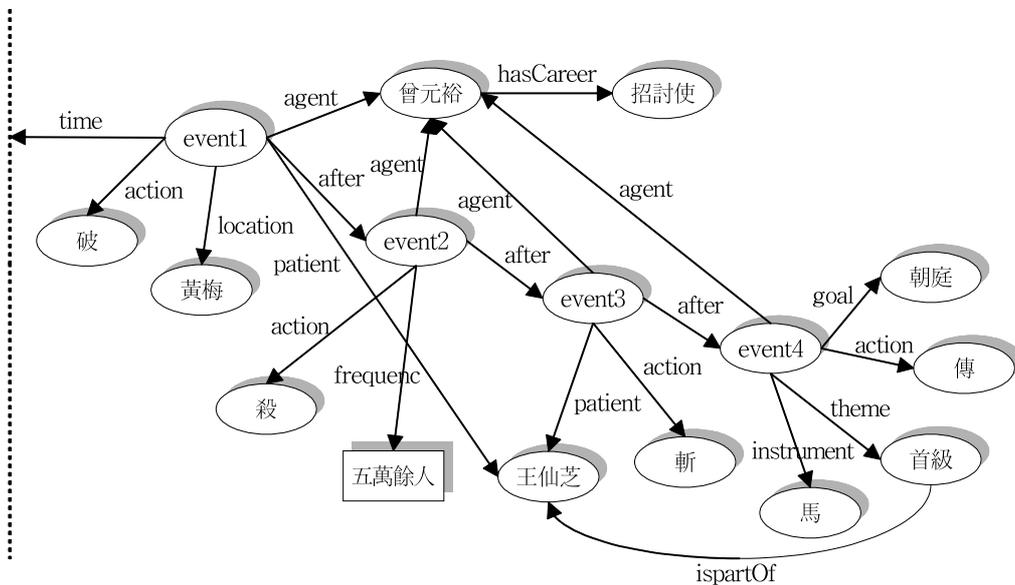


그림 75. 파발마를 도구로 표현

(11) 피동주와 대상

RDF 트리플이 주어-술어-목적어 관계로 표현하기 때문에 목적어 위치에 올 수 있는 것이 우리말의 목적어로 이해하기 쉬우나 반드시 그런 것은 아니다. 아래의 RDF 그래프에서 문법적으로 목적어에 해당하는 것은 ‘patient’와 ‘theme’이나, RDF 트리플 관점으로 보면 모든 eventRole로 연결되는 모든 개체가 목적어가 된다.

- 明年 賜叔文死
- ☞ 다음 해[time]에 王叔文[patient]에게 사약[theme]을 하사하다[paction]

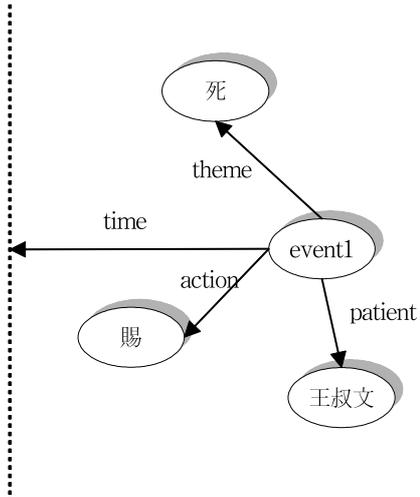


그림 76. 王叔文에게 사약을 하사하दा를 표현한 그래프

(12) 임면

역사 자료에서 관작을 任命하고 罷職하는 내용은 빈번하게 나타난다. 아래 문장에서 皇帝는 ‘唐昭宗 李曄’이며, 李克用은 後唐 莊宗 李存勖의 아버지이다. 唐昭宗이 李克用의 관작을 빼앗은 후에 張濬을 河東行營招討制置宣慰使로 임명한다. 이때 李克用에게서 삭탈한 관직이 ‘河東行營招討制置宣慰使’인지 정확하게 드러나지 않았으므로 일단은 ‘theme’를 관직으로 한다. 그리고 관작을 임명한 결과는 goal로 표현한다. event1과 event2는 선후 관계가 있어 ‘after’로 표현하였다.

■ 五月 詔 削奪克用官爵 以濬 爲河東行營招討 制置宣慰使

☞ 5월에 황제가 명하여 李克用의 관작을 삭탈하고, 張濬을 河東行營招討 制置宣慰使로 임명하였다.

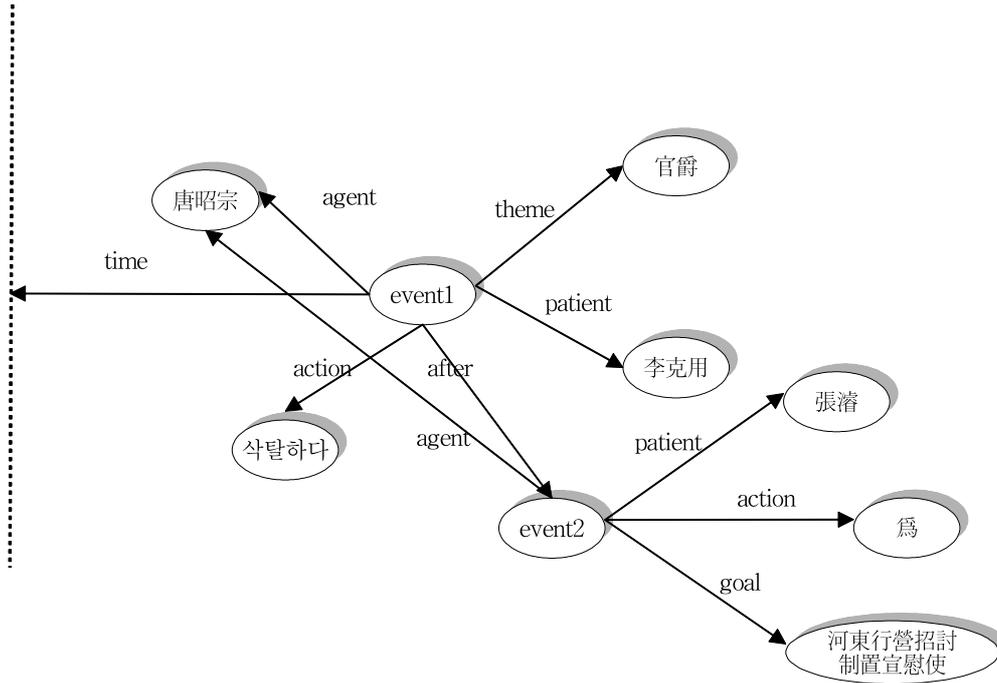


그림 77. 관직을 삭탈함을 표현하는 그래프

■ 詔以行瑜 爲靜難節度使

☞ 황제가 명하여 王行瑜를 靜難節度使로 임명하였다.

■ 十月 下詔 廢王皇后, 蕭淑妃 爲庶人 命司空李勣 齎璽綬 冊皇后武氏

☞ 10월에 上이 詔命을 내려 王皇后와 蕭淑妃를 폐하여 서인으로 삼고, 司空 李勣에게 명하여 옥새와 인끈을 가지고 가서 武氏를 황후로 책봉하게 하였다.

위의 문장에서 ‘하게 하였다’가 서인으로 삼고 황후로 책봉한 두 가지 행위에 모두 연결이 된다. 이러한 유형으로는 ‘~에 하게 하다, ~에 임명하다, ~로 봉하다, ~에 봉하다’ 등이 있다.

■ 周主以是賢之 卽位 首用爲相

☞ 周主가 이 때문에 그를 어질게 여겨서 즉위한 뒤에 첫 번째로 등용하여 재상으로 임명하였다.

아래 문장들도 마찬가지로 goal로 관직의 임명을 표현한 예이다.

■ 八月庚子에 制호되 令太子로 卽皇帝位하고 朕稱太上皇이라하고 徙居興慶宮

☞ 황제[agent]가 8월 庚子日(4일)[time]에 興慶宮[goal]으로 거처[theme]를 옮기다[action]

- 以王伾爲左散騎常侍 王叔文爲起居舍人
 - ☞ 황제[agent]가 王伾[patinet]를 開州司馬[goal]로 좌천시키다[action]
- 貶王伾開州司馬 王叔文渝州司戶
 - ☞ 王叔文[patient]을 渝州司戶[goal]로 좌천시키다[action]

임면에 주로 사용하는 goal은 이동을 나타내는 동사의 목적지로도 많이 사용된다.

- 庚寅旦 袁象先 帥禁兵數千人 突入宮中
 - ☞ 庚寅日(2월 17일) 새벽에 袁象先이 禁兵 수천 명을 거느리고 궁중으로 돌입하니

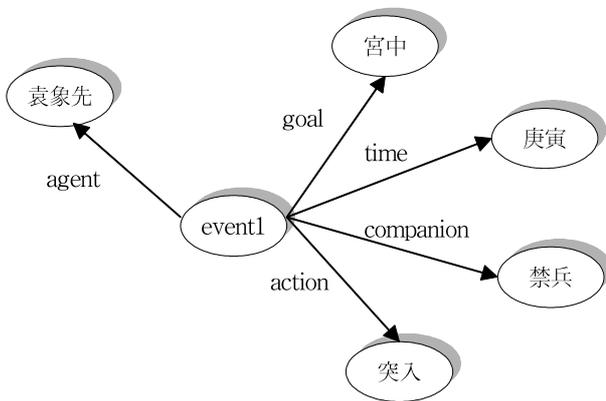


그림 78. goal을 이용한 목적지의 표현

(13) 요청

아래 문장에는 의도와 요청이 모두 포함되어 있다. ‘太原으로 가려 하였는데’는 의도이며 ‘華州로 갈 것을 청하니’는 요청이다. 황제[唐昭宗]가 의도한 내용은 event1인 ‘太原으로 가려 하다’이며, event2에서 event1을 theme로 연결하였다. event6의 행동주인 韓建이 요청한 내용은 ‘event5’가 되고 ‘event5’의 내용은 ‘華州로 가다’이다. event4의 ‘당 소종이 ‘華州로 가다’는 요청을 따랐다는 내용은 다시 event5가 된다. event 간의 관계가 복잡하지만 표현하는 데는 문제가 없다.

- 李茂貞 犯京師 帝將幸太原 韓建 請幸華州 請幸華州 遂欲制之 上從之
 - ☞ 李茂貞이 京師를 침범하자 황제가 장차 太原으로 가려 하였는데, 韓建이 華州로 갈 것을 청하니, 上이 그의 말을 따랐다.

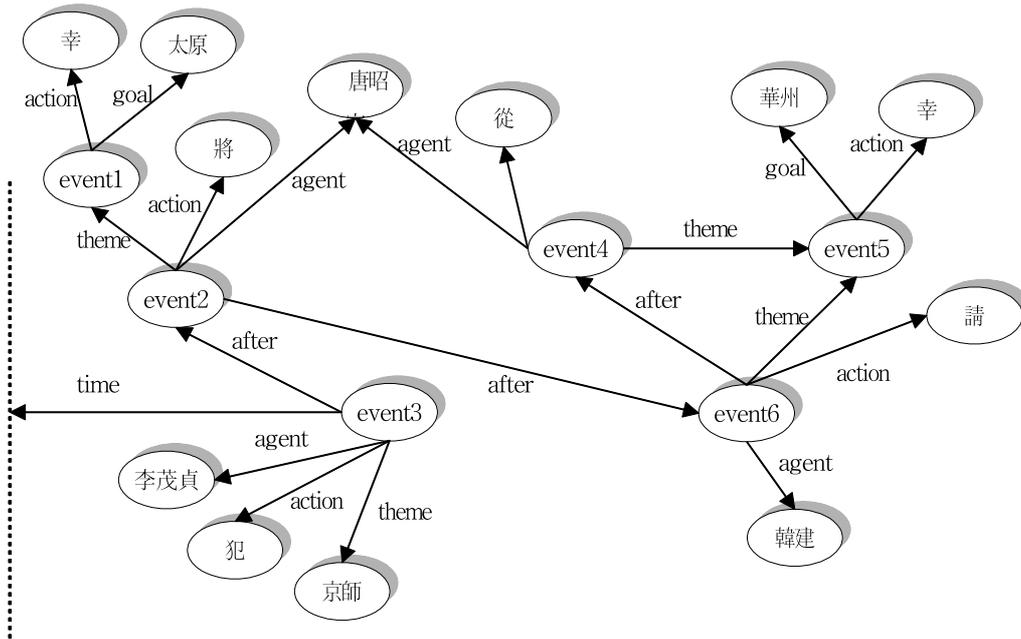


그림 79. 요청하는 내용을 표현한 그래프

나) 동사의 대역어 표현

기본적으로 漢文과 漢文을 번역한 翻譯文에 사용되는 어휘를 비교해 보면 명사는 대부분 翻譯文에서도 그대로 나타난다. 그렇지만 동사나 형용사의 경우 漢字語에 ‘하다’를 붙여서 번역하거나 아니면 文脈에 따라 적절한 우리말로 번역하기도 한다.³⁶³⁾ 한문 문장을 번역문을 이용하여 意味役에 따라 표현할 때 가장 문제가 되는 것은 보조 용언의 표현이다. 예를 들어 한국어에서 보조적 연결 어미 ‘-아/-어, -게, -지, -고’ 등을 매개로 하여 본용언에 연결하는 보조 용언을 RDF 트리플로 표현하면 RDF 트리플 문장이 복잡해지는 등의 문제가 있다.

아래 문장에서 ‘中使를 보내 호송하였다’는 호송하도록 명하거나 시킨 것으로 볼 수 있으며, 그 행위의 주체는 ‘盧杞’이고 피동주는 ‘楊炎’이 된다.

■ 盧杞譖楊炎 十月 貶崖州司馬 遣中使護送 未至崖州 縊殺之

☞ 盧杞가 楊炎을 참소하여 10월에 崖州司馬로 좌천시키고 中使를 보내어 호송하였는데, 崖

363) 전통문화연구회의 번역 방식은 가급적 한자어를 살려주면서 한자를 그대로 노출시키는 것을 원칙으로 하고 있다.

州에 이르기 전에 그를 목졸라 죽였다.

이를 RDF 그래프로 표현하면 아래와 같다.

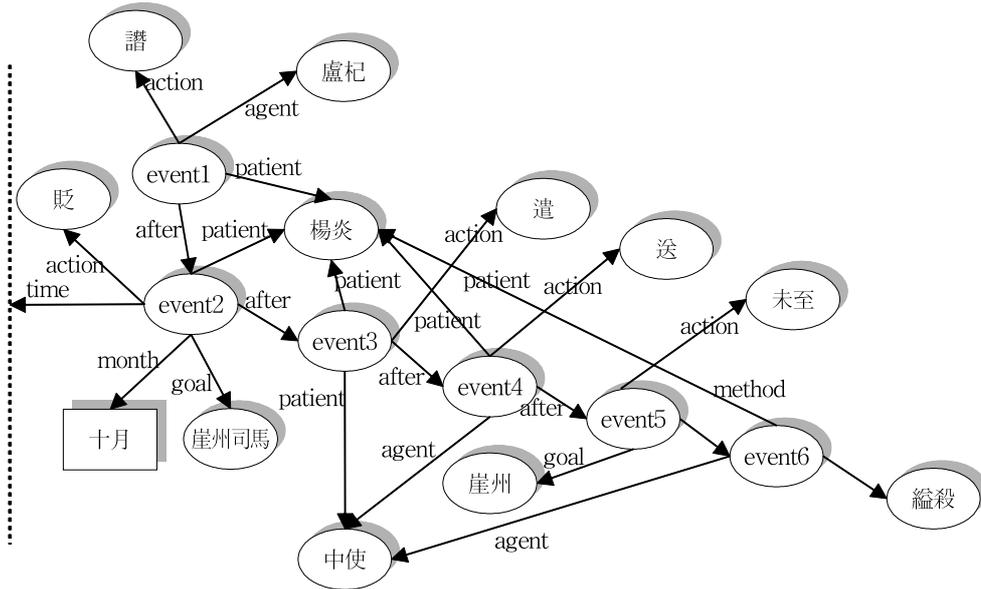


그림 80. 『通鑑節要』를 대상으로 RDF 트리플 표현

다음은 위의 RDF 트리플을 우리말로 표현한 것이다.

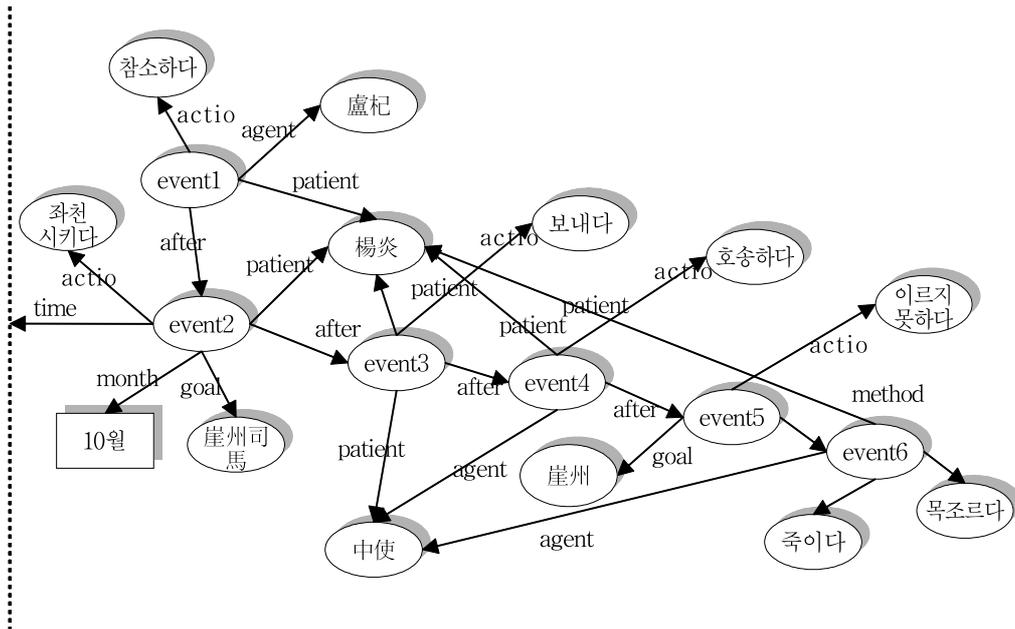


그림 81. 『通鑑節要』 번역문을 대상으로 RDF 트리플을 표현한 예

번역문을 이용하여 RDF 트리플을 표현하면 ‘이르지 못하다’와 같이 부정문의 표현이 길어지는 등의 문제가 있으나 RDF 트리플로 표현하는 데 큰 무리는 없다. 이들 원문과 번역문에 사용된 모든 어휘는 1:1로 매핑을 하여 자동 번역 등에 활용할 수 있도록 한다.

4) 인터링킹

인터링킹은 서로 다른 데이터 정보원들 사이에 데이터를 연결하는 것을 말한다. 국립중앙도서관에서는 국가서지 링크드 데이터를 구축하고 외부 LOD와 상호연계를 확대하기 위해 국내에서는 KDATA, 한국사 LOD, 한국교육학술정보원(RISS) LOD, 생물정보LOD와 인터링킹하고, 국외 LOD는 OCLC WorldCat, 독일 국립도서관, 영국 국립도서관, 미의회도서관 등과 인터링킹하였다.³⁶⁴⁾ ‘owl:sameAs’는 두 개의 자원이 같을 경우에 사용하여 연결한다고 하였다. 그렇다면 같다는 의미가 무엇인지 살펴보자. 같고 다름을 판단할 때 개념에 대해서 정확하게 이해해야 한다. 여기서 개념을 말하는 이유는 온톨로지가 개념을 대상으로 하고 있기 때문이다. 온톨로지의 설계 시에 개념과 어휘 혹은 실체를 嚴格하게 구별해야 한다. 개념은 세 가지 성질을 가지며 思考의 형식화를 위해서 사용된다. 세 가지 성질이란 첫째,

364) 이현주, 「학술연구 : 국립중앙도서관 국가서지 LOD 구축 사례」, 『디지털도서관』, 제77권, 2015, 27쪽.

개념은 말로 나타낸다. 둘째, 개념은 인식의 대상이 되는 사물의 集合(外延)을 규정한다. 셋째 개념은 그 집합이 공통으로 가지고 있는 性質(內包)을 규정한다.³⁶⁵⁾

다음의 문장을 보자. 여기서 釋義는 『通鑑釋義』를 줄인말이며 明나라의 王逢이 주석을 단 책을 말한다.

王 命掌書記王緘 草露布[釋義] 露布文心雕龍曰 露布者 蓋露板不封 布諸視聽也

위의 문장에서 露布를 설명하기 위해서 『文心雕龍』에 있는 “露布는 露板을 봉합하지 않아서 여러 사람이 보고 듣게 하는 것이다.”라는 내용을 인용하여 설명하였다. 여기서의 『文心雕龍』은 구체적인 실체를 의미하는 것이 아니라, 內包하고 있는 의미는 ‘중국 梁나라의 劉勰이 쓴 남북조 시대의 문학 평론서’를 말한다. 『文心雕龍』이라는 책이 內包하고 있는 특성을 가진 구체적인 對象을 보면 『通鑑節要』를 편찬할 때 참조한 『文心雕龍』이라는 책이 있을 것이고, 현재 국립중앙도서관에 소장되어 있는 청구기호가 ‘古古5-60-13’인 『文心雕龍』이라는 책도 있을 것이며,³⁶⁶⁾ 현대에 간행된 『文心雕龍』이라는 책도 있을 것이다. 그렇다면 『通鑑節要』에서 언급한 『文心雕龍』과 국립중앙도서관에 소장되어 있는 『文心雕龍』을 같다고 할 수 있을까? 만약 본문 중에서 『文心雕龍』이 어떤 특정한 책을 지칭했다면 그것은 하나의 실체라고 볼 수 있다. 그렇지 않다면 그것은 개념적인 것을 말하며, 특정한 물리적인 형태를 지니고 있는 어떤 특정한 책을 말하는 것이 아니다. 즉 본문 중에 출현하는 『文心雕龍』과 국립중앙도서관에서 소장되어 있는 『文心雕龍』은 서로 같지 않다는 것이다. 그러므로 본문 중에 나오는 『文心雕龍』과 국립중앙도서관에서 발행한 LOD는 owl:sameAs를 이용하여 연결할 수는 없는 것이다. 만약 국립중앙도서관에서 발행한 LOD가 文心雕龍 자체를 의미하는 것이라면 이는 연결할 수 있을 것이다. 그렇다면 WIKIDATA에 등재되어 있는 文心雕龍을 보자. 아래 그림에서 보듯이 WIKIDATA에는 각 국가별 10개의 엔티티가 있음을 보여주고 있다.³⁶⁷⁾ 이들 콘텐츠는 개별적인 文心雕龍이라는 책을 설명한 것이 아니라 文心雕龍이라는 책의 개념에 대해서 설명하고 있다. 그러므로 이러한 자료에는 owl:sameAs로 연결하는 것이 적절하다고 할 수 있다.

365) 齋藤孝(저)/최석두·한상길(역), 『온톨로지 알고리즘』1, 한울아카데미, 2008, 117쪽.

366) 이 책은 구체적인 실체를 가지고 있으며 이 책은 LOD로 발행할 때

‘<http://lod.nl.go.kr/page/EOL000002052>’라는 URI 주소를 부여하였다.

367) WIKIDATA, <https://www.wikidata.org/wiki/Q1153676>(2017. 7. 8)

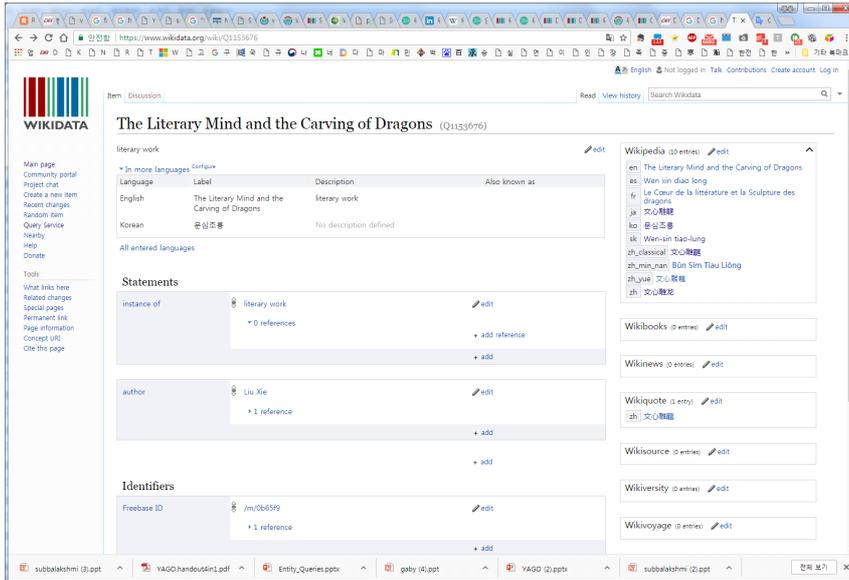


그림 82. WIKIDATA에서 제공하는 문심조룡 메타데이터

5) LOD 등록 및 공개

LOD로 발행한 뒤에는 가급적 많은 사람들이 데이터를 이용할 수 있도록 공개해야 한다. LOD(Linked Open Data)는 데이터를 웹에 공개, 공유할 수 있으며 URI를 통해 데이터 이용이 가능하다. 구축된 LOD의 경우는 인터넷상에 공개되기 때문에 수동적으로 웹 크롤러에 의하여 수집되어 공개되기를 바랄 수도 있고, 능동적으로 LOD를 공유하는 DataHubData Hub³⁶⁸⁾사이트에 직접 등록할 수도 있다.³⁶⁹⁾ 데이터를 공유하기 위한 과정은 'https://datahub.io/en/about'에서 확인할 수 있다.

다. 활용 방안

1) 시각화를 통한 이벤트의 표현

구축한 온톨로지를 일반인들이 가장 쉽게 이해할 수 있는 것은 개체와 개체의 관계를 시각적으로 표현하는 방법일 것이다. 노드의 관계를 시각적으로 표현하는 방법에는 여러 가지가 있으나 본 논문에서는 Neo4j를 활용하여 시각화를 시도해 본다. Neo4j는³⁷⁰⁾ Java 언어

368) datahub, <https://datahub.io/dataset>

369) 황미영 외, 앞의 논문(제39권, 제1호, 2012), 94쪽.

를 이용하여 개발된 그래프 DB로 Oracle이나 Mysql처럼 data를 저장하고 관리해 주는 NoSQL DBMS 중의 하나이다. Neo4j는 노드(node), 엣지(edge)로 구성된 그래프의 형태로 데이터를 저장하고 관리한다. community 버전은 무료로 사용할 수 있으나 웹 서비스를 위해서는 상용 버전을 구매해야 한다. Neo4j에서 RDF 그래프를 시각화하기 위해서는 RDF 트리플을 Neo4j 형식에 맞게 변환해야 한다. RDF 트리플이 노드와 노드의 관계를 표현한 것이기 때문에 이론적으로는 Neo4j 형식으로 변환하는 것은 큰 문제가 없어야 한다. RDF를 Neo4j 포맷으로 변환해 주는 오픈소스 도구가 있으나³⁷¹⁾ 본 논문에서는 엑셀을 이용하여 노드와 릴레이션을 만든 후에 Neo4j에 업로드 하는 방식을 사용하였다.

노드와 노드 간의 관계를 도형과 화살표 등을 이용하여 시각적으로 표현한다는 것은 정보를 직관적으로 이해할 수 있다는 점에서 분명 매력적인 일임에는 틀림없으나 모든 관계에 유용한 것은 아니다. 역사 자료에서 나타나는 관계를 살펴보면 앞서 언급한 것처럼 정적인 관계와 동적인 관계로 구분할 수 있다. 사람 간의 관계, 사람과 지명의 관계는 정적인 관계이기 때문에 시각적으로 표현하는 데 큰 문제가 없다. 그러나 노드 간에 발생한 행위를 표현하는 것은 사정이 좀 다르다. 왜냐하면 행위에는 시간이란 요소가 추가되기 때문이다. 노드 간에 발생한 행위 이벤트는 노드와 노드 간의 관계로 표현할 수 있지만 이들 간의 관계는 시간의 흐름에 따라 여러 번 발생할 수 있기 때문이다. 특히 본 논문에서 역사적 지식을 LOD로 구축하는 방법으로 event와 개체의 관계를 意味役으로 표현하였기 때문에 논리적인 추론은 SPARQL 등을 이용하여 가능하지만, 시각적으로 표현하는 것은 직접 개체와 개체의 관계가 연결이 되지 않기 때문에 직접 연결되지 않는 많은 노드가 생겨날 수 있다. 역사서에 출현하는 사건은 단순히 정적인 관계뿐만 아니라 동적인 관계도 있기 때문에 행위를 표현할 수 있어야 한다. 이를 위해 중국 역사 중 극심한 혼란기였던 오대십국의 역사를 통해서 구체적으로 행위 온톨로지를 표현하는 방법을 찾아보기로 한다.

아래 그림은 五代十國 당시 5대로 일컬어지는 ‘後梁·後唐·後晉·後漢·後周’ 등 다섯나라의 혈연 관계나 사회적 관계를 그림으로 표현한 것이다. 혈연 관계와 혼인 관계 및 사회적 관계를 정확하게 알면 역사를 바라보는 시각이 달라지고, 사료를 해석하는 맛도 달라진다.³⁷²⁾ 중국 역사상 극심한 혼란기였던 오대십국의 역사를 살펴보면 형제 간에 서로 죽고 죽이는 다툼이 여러 번 발견된다. 신하가 황제에게 즉위하도록 부추기거나 황제를 죽이는 일도 빈

370) Neo Technology, Inc., <https://neo4j.com>

371) Importing RDF data into Neo4j,

<https://jesusbarrasa.wordpress.com/2016/06/07/importing-rdf-data-into-neo4j> 이 사이트에서는

RDF 그래프 데이터를 Neo4j로 데이터베이스로 마이그레이션하는 방법을 설명한다. 지원하는

포맷은 JSON-LD, Turtle, RDF/XML, N-Triples 및 TriG이다.

372) 이성무 외, 『조선의 이끈 명문가 지도』, 글항아리, 2011, 13쪽.

번하게 발생한다. 이러한 시기에 역사적 사실이 인물 관계와 행위를 통해서 구체화된다면 역사를 이해하는데 많은 도움이 될 것이다.

오대십국은 907년 黃巢의 부장이었던 朱全忠이 唐 哀帝로부터 禪讓의 형식으로 제위를 물려 받아 後梁을 건국한 이후부터, 중원 지역에서 後梁·後唐·後晉·後漢·後周의 다섯 왕조가 이어지고, 그 외의 지역에서 南吳·南唐·吳越·閩·荊南·楚·南漢·前蜀·後蜀·北漢의 10국이 할거했던 시기를 말한다. 오대십국은 960년 宋 太祖 趙匡胤이 後周의 恭帝로부터 제위를 禪讓받음으로써 끝난다.³⁷³⁾ 907년부터 960년까지 약 50년이 조금 넘는 기간이었으나 이 기간 동안 형제 간의 살육과 군신 간의 다툼이 빈번했던 시기였다. 이 시기를 황제를 중심으로 사람과 사람의 관계를 사건 중심으로 표현하면 다음의 그림과 같다. 기본적인 관계는 형제 관계와 군신 관계 등으로 표현하였다. 그림을 알 수 있듯이 唐나라 僖宗 때 王仙芝가 처음 당나라의 반기를 든 이후에 黃巢가 천 여명의 무리를 모아 王仙芝에 호응하였다. 朱全忠은 원래 이름이 朱溫으로 黃巢가 同州防禦使로 임명하였으나 시세를 보고 唐나라에 귀순하여 당으로부터 朱全忠이라는 이름을 하사받았다. 朱全忠은 뒤에 蔣玄暉를 시켜 당 昭宗을 시해하고 哀帝를 황제로 세운다. 哀帝는 제위가 3년이고 朱全忠에게 禪讓이라는 형식을 빌어서 제위를 넘기게 된다. 이에 朱全忠은 後梁이란 이름으로 새로운 왕조를 건국하게 된다. 朱全忠은 養子인 朱友文의 妻를 사랑하여 朱友文을 후사로 세우려고 하다가 마침내 朱友珪에게 시해당한다. 朱友珪는 梁나라 太祖의 次子이니, 그의 어머니는 亳州營의 娼妓였다. 얼마 후 朱全忠의 3남인 朱友貞이 朱友珪를 살해하고 황제에 오른다. 朱全忠과 晉王 李克用은 서로 경쟁하는 관계였었는데, 李克用이 죽자 李克用的 아들 李存勖이 後梁의 朱友貞을 공격하여 大梁을 함락시킨다. 朱友貞은 살아날 수 없음을 알고 측근인 皇甫麟에게 자신을 살해할 것을 지시한다. 이에 皇甫麟은 황제를 죽이고 자신도 스스로 죽는다. 이로서 後梁은 멸망하게 된다.

李嗣源는 李克用的 양자로서 李存勖이 술에 빠져 방탕하자 제장들의 추대를 받아 황제로 즉위한다. 李嗣源의 죽자 삼남인 李從厚가 즉위하였으나, 그가 어리석어서 대신인 朱弘昭와 馮贇에게 좌지우지되자 李嗣源의 양자인 李從珂가 군사를 일으켜 李從厚를 폐하고 황제에 오른다. 후에 李從珂는 李從厚를 살해한다. 石敬瑭은 李嗣源의 사위인데, 李從珂가 황제에 오른 뒤 石敬瑭을 두려워하였다. 李從珂는 石敬瑭을 제거하기 위해 石敬瑭을 鄆州로 옮기려 하였는데 石敬瑭은 劉知遠의 권유에 의해 자립하여, 後晉을 건국한다. 石敬瑭은 遼나라 군대를 동원하여 李從珂가 있는 洛陽을 포위하였고, 李從珂는 玄武樓에 올라가 스스로 불타 죽게 된다. 石敬瑭이 죽은 후 조카인 石重貴가 황제가 되자 石重貴는 劉知遠을 의심하게 된다. 이에 石重貴는 劉知遠의 藩鎮을 옮기도록 한다. 이때 劉知遠의 신하였던 郭威가 자립을

373) 五代十國 시대의 역사는 그렇게 끝나는 것으로 되어 있으나 실제로는 宋나라가 건국한 뒤에도 상당 시간 다른 王朝가 유지되고 있었다.

권하자 劉知遠은 독립을 하여 後漢을 건국한다. 後漢의 劉知遠이 죽자 아들인 劉承祐가 즉위한다. 이때 後漢에서 楊邠은 機務와 정사를 총괄하고, 郭威는 정벌을 주관하고, 史弘肇는 宿衛를 맡고, 王章은 財賦를 맡아서 국가가 안정되고 있었는데, 劉承祐가 장성하자 大臣들에게 제재받는 것을 싫어 하여, 楊邠, 史弘肇, 王章을 죽이고, 孟業을 보내 郭威를 죽이도록 한다. 이에 郭威가 제장들을 이끌고 劉承祐를 공격하자, 劉承祐는 전란 중에 시해당한다. 郭威는 劉贇을 세우고 얼마 후에 직접 황제로 즉위하여 後周를 건국하고 劉贇을 살해한다. 後周의 郭威가 죽자 조카인 郭榮이 황제가 되었는데 이가 五代十國 시대의 제일 명군으로 알려진 後周 世宗이다. 郭榮이 중국의 통일을 위해 정벌을 하던 중 죽게 되자 아들 柴宗訓이 7살의 나이로 제위를 물려받게 된다. 柴宗訓은 즉위한 지 1년만에 宋 太祖인 趙匡胤에게 禪讓하게 된다.

다음 그림은 王仙芝가 반란을 일으킨 시점으로부터 宋 太祖가 중국을 통일하기 전까지의 기간 동안에 각국 황제들 간의 혈연 관계와 사회적 관계를 표현한 것이다. 편의상 관계를 사각형으로 표현하였다. 위의 내용을 근거로 살펴보면 각 왕조 간의 관계를 대략적으로 파악할 수 있다.

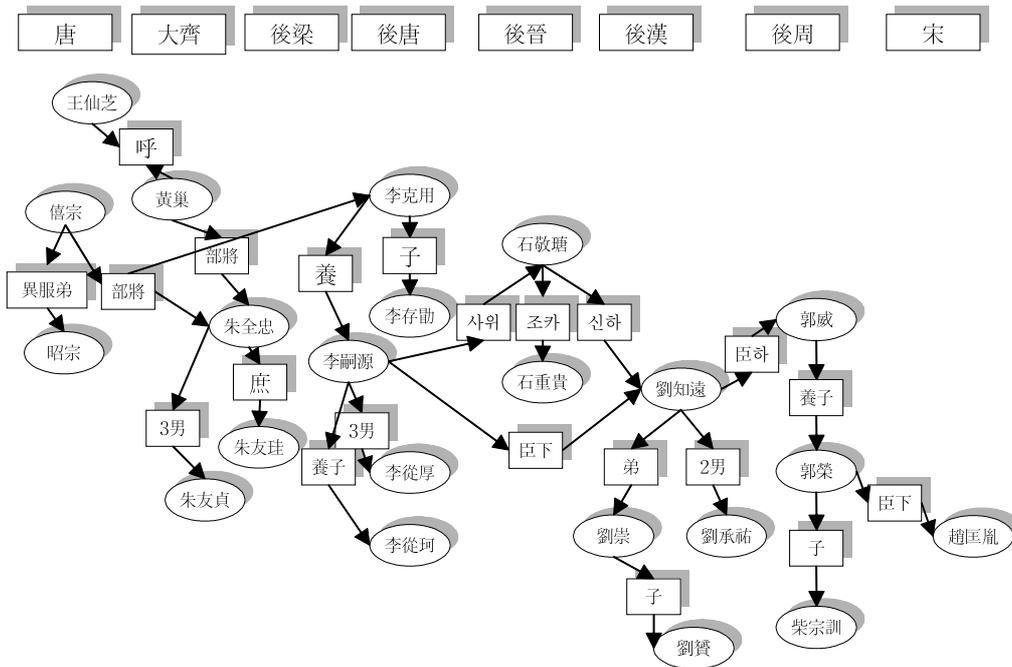


그림 83. 五대의 인물간 관계

그러나 역사적 사실을 단순히 관계에 의해서만 표현할 수는 없다. 이벤트 간에 어떤 행위

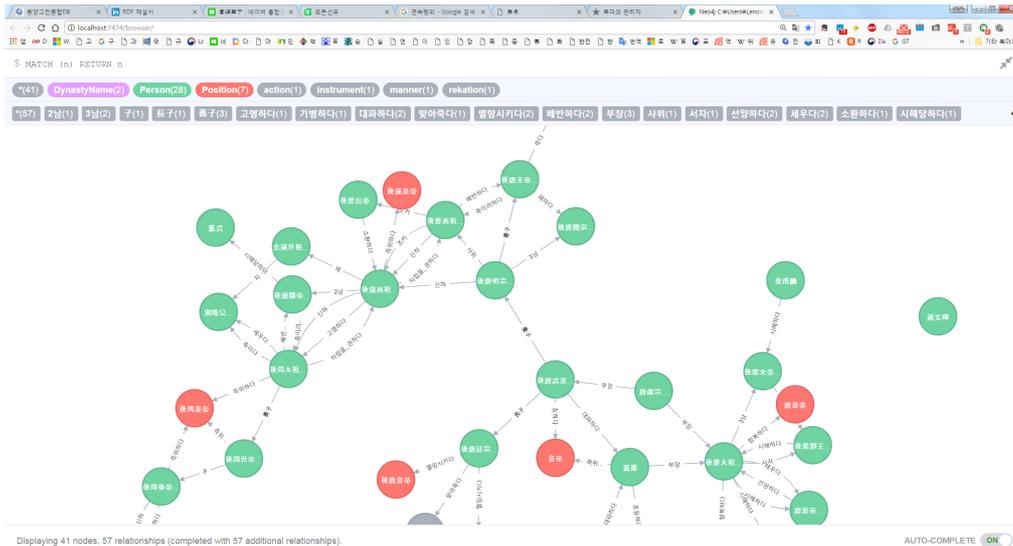


그림 85. Neo4J를 이용하여 event를 시각화한 결과

2) 온톨로지를 이용한 지식의 표현

지금까지 의미역을 이용하여 지식을 표현하는 방법에 대해 언급하였는데 이를 辭典에 확장해서 적용해 본다. 사전에는 정적인 관계와 동적인 관계가 모두 존재하며, 지식의 결정체라는 점 때문에 사전을 추론 가능한 지식으로 변환한다면 그 활용 분야는 매우 크다고 할 수 있다. 辭典 내용을 RDF로 구축하기 위해서는 어휘의 정의를 표현할 수 있어야 하며, 사전의 풀이말에 나타나는 어휘 간의 다양한 관계와 정보를 표현할 수 있어야 한다.

먼저 辭典의 구조를 간단히 살펴 보자. 먼저 표제어는 『漢韓大辭典』에 나오는 ‘侯進忠’이라는 사람이다. 이 표제어의 성격에 따라 관계의 표현이 달라진다. 아래의 경우 표제어가 사람이기 때문에 본문 중에 나타나는 내용은 대부분 사람을 중심으로 관계를 표현할 수 있다. 예를 들어 ‘侯進忠’의 國籍이나 字, 官職 등에 대해서 설명하였다.

<표제어>侯進忠</표제어>
 <풀이말><nation type="국가" rel="국적">명(明)</nation> <place type="지명" rel="출신지">봉양(鳳陽)</place> 사람. 자는 <person type="자" rel="자">단일(丹一)</person>. <position rel="관직">천호(千戶)</position>를 세습하여 <position rel="관직">유격(游擊)</position>으로 승진하였다. 그림과 초서(草書)에 뛰어나고 활을 잘 쏘았다.</풀이말>

- 侯進忠 - 字 - 丹一

- 侯進忠 - 官職은 - 千戶
- 侯進忠 - 세습하다 - 千戶
- 侯進忠 - 官職은 - 遊擊
- 侯進忠 - 승진하다 - 遊擊
- 侯進忠 - 뛰어난다 - 그림
- 侯進忠 - 뛰어난다 - 草書
- 侯進忠 - 國籍 - 明
- 侯進忠 - 出身地 - 鳳陽

이를 RDF 그래프로 표현하면 다음과 같다.

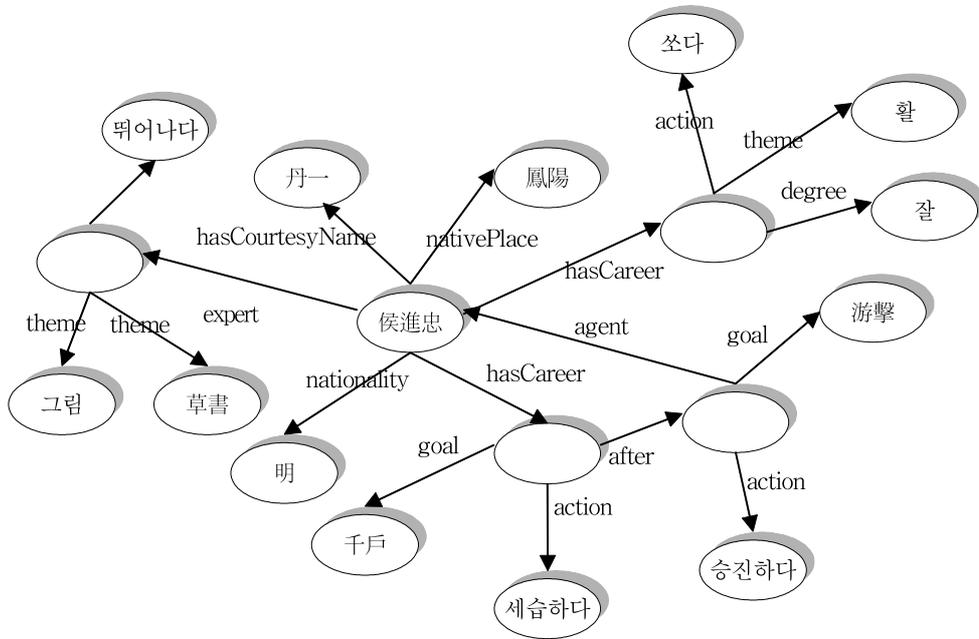


그림 86. 인물의 설명문을 대상으로 RDF 그래프로 표현

다음은 一家言이라는 책에 대한 설명이다.

<표제어>一家言</표제어>
 <풀이말>청(淸) 이어(李漁)의 시문(詩文)과 잡저(雜著)를 모아 엮은 책. 6권. 李笠翁一家言.</풀이말>

- 李漁의 저서는 一家言이다.

- 李漁의 국적은 淸이다.
- 一家言의 권차는 6권이다.(별도의 주어가 표시되지 않은 것은 표제어가 주어가 된다.)
- 一家言의 이칭은 李笠翁一家言이다.

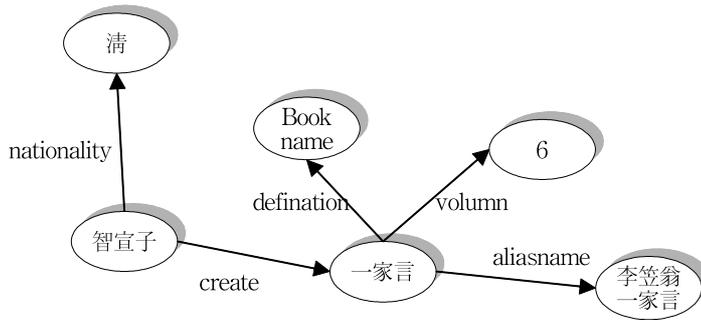


그림 87. 서명의 설명문을 RDF 그래프로 표현

사전의 경우 그 내용이 간결하기 때문에 일반 역사서에 비해서 RDF로 변환하는 것이 더 용이할 것으로 보인다.

라. 시맨틱 질의를 통한 실험 및 평가

지금까지 연구한 온톨로지 설계를 바탕으로 LOD를 발행하기 위해 『通鑑節要』 내에 있는 인물과 지명 등의 인스턴스와 이들 간의 상호 관계, 행위와 상태를 중심으로 한 event를 RDF 트리플로 구축하였으며, 연구자들이 직접 그 결과를 이용할 수 있도록 버추오소 (Virtuoso) 프로그램을 이용하여 추출한 RDF 트리플을 스토리지에 저장하고 LOD로 발행하였다. 또한 『통감절요』 본문에서 나타나지 않는 연표 등의 기초 지식을 사전 등을 참조로 하여 지식 관계망을 시험적으로 구축하였다.

구체적으로 唐나라가 쇠하기 시작한 宣宗부터 後周의 柴宗訓이 宋 太祖에게 선양할 때까지의 시기이며, 분량으로는 전체 50권 중 48권부터 50권까지 3권에 해당한다. 또한 『通鑑節要』에서는 나오지 않지만 역사를 이해하는 데 꼭 필요한 기초 자료인 연표, 황제 가계, 황제의 자손 가계는 중문 위키피디아를 이용하여 정리하였다.

표 87. LOD 구축량

구분	내용	비고
시대	唐宣宗~柴宗訓	
대상 자료	『通鑑節要』 卷48, 49, 50	전체 50권 중 3권
사건 이벤트	1,787건	

다음으로 몇 가지 사용자 질의 시나리오를 통해 일반적인 검색으로 조회할 수 없었던 새로운 지식을 탐색할 수 있음을 보여줌으로써 온톨로지 설계가 의미가 있음을 증명한다. 통감절요 RDF 데이터 셋은 SPARQL(Simple Protocol and RDF Query Language)을 이용하여 탐색해야 한다. SPARQL은 RDF 형식의 데이터를 탐색할 수 있는 데이터베이스 질의어이다. W3C에서 2008년에 처음 제안하였다. 2013년 이후에는 SPARQL 1.1이 W3C의 공식 권장안이다.³⁷⁴⁾ SPARQL을 이용하여 구체적인 결과를 보여주는 실험은 Virtuoso에서 제공하는 Virtuoso SPARQL Query Editor³⁷⁵⁾을 이용한다. LOD에서 모든 데이터(LOD, Triple, HTML 등)는 SPARQL Endpoint를 통해 유통된다. LOD는 브라우저를 통해 URI에 접근하기도 하지만 대부분 기계가 이해하기 위한 목적으로 만들어지며, 기계적으로 LOD에 질의하는 경우에 대응하기 위한 인터페이스가 SPARQL Endpoint이다. SPARQL Endpoint는 사용자로부터 RDF 질의문인 SPARQL을 입력 받아, 질의문에 적합한 결과를 보여준다. 그러나 이 방법은 사용자가 직접 질의문을 작성하여 결과를 얻기 때문에 사용자가 SPARQL 질의문에 대한 지식이 있어야 한다는 단점이 있다.

SPARQL로 검색을 하기 위해서는 LOD 데이터의 구조를 파악해야 하며, 그 구조는 온톨로지 설계에서 확인해야 한다. 그리고 찾고자 하는 대상이 무엇인지를 파악해야 한다. 찾고자 하는 요소가 클래스인지 프로퍼티인지를 확인하고, 그들 간의 관계를 어떤 용어를 사용했는지 확인해야 한다. 또한 찾고자 하는 대상이 주어인지 술어인지 목적어인지를 정확하게 파악해야 한다.

SPARQL은 아래와 같은 SELECT, CONSTRUCT, ASK, DESCRIBE 등의 유형으로 구분할 수 있다.

374) 김현 외, 『디지털 인문학 입문』, 한국외국어대학교 지식출판원, 2016, 181쪽.

375) <http://61.35.170.118:8890/sparql>

표 88. SPARQL의 유형³⁷⁶⁾

질의 형식	용도	결과
SELECT	RDF 그래프로부터 주어진 조건과 일치하는 노드를 찾거나 병합하거나 한다.	변수와 이에 매치된 값
ASK	where 절에 작성한 트리플 패턴이 대답과 일치하는지 확인하는 기능을 제공한다.	부울값
DESCRIBE	질의는 매개변수로 하나의 URI를 사용하며, 주어진 자원을 설명하는 RDF 그래프를 반환한다.	RDF 그래프
CONSTRUCT	DESCRIBE 질의처럼 RDF 그래프를 반환한다. WHERE 절의 질의 정보로 새로운 RDF를 생성한다.	RDF 그래프

SPARQL은 다음과 내용을 포함한다.

1. PREFIX(네임스페이스 접두어)
2. SELECT(조회할 내용을 정의)
3. FROM(조회한 결과를 얻기 위한 데이터 세트를 정의)
4. WHERE(RDF 트리플 형태로 조회할 조건을 정의)
5. ORDER BY, LIMIT 등(결과의 순서, 제한 등을 정의)

먼저 기본적인 SPARQL 구문을 살펴보자.

표 89. 기본적인 SPARQL 구문

기본 형태	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> select * where { ?htimetemp rdf:type kno:Period . } </pre>
축약 형태	<pre> select * where { ?htimetemp a <http://dh.aks.ac.kr/ontologies/historicevent/kno#Period> . } </pre>

SPARQL에서 모든 자원은 URI를 이용하여 표현하며 모든 URI는 ‘< >’ 안에 넣는다. OWL에서 자료의 유형을 나타낼 때 ‘rdf:type’³⁷⁷⁾을 사용한다. PREFIX는 URI를 간결하게 줄이기 위한 장치로 PREFIX를 선언하고 그 뒤에 ‘kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#>’처럼 축소한 이름과 전체 URI 주소를 기재하면 SPARQL 구문에

376) David Wood 외(저)/오원석 외(역), 앞의 책(2014), 226쪽.

377) rdf(<https://www.w3.org/1999/02/22-rdf-syntax-ns>)에 정의되어 있는 속성이다.

서 'kno:'만 사용해서 '<http://dh.aks.ac.kr/ontologies/historicevent/kno#>'를 대체할 수 있다. SPARQL 쿼리는 기본적으로 RDF 트리플 패턴으로 하여 찾고자 하는 결과를 조회한다. WHERE 조건에 이 주어부(subject), 술어부(predicate), 목적부(object) 패턴을 제시함으로써 조건에 맞는 결과를 뽑아낼 수 있다. 이 주어-술어-목적어 패턴은 여러 번 나열할 수 있다. 이 패턴은 한 줄에 모두 써 주어도 되나 가독성을 고려하여 행을 바꿔서 써 주는 것이 일반적이다. 이때 주어-술어-목적어를 하나의 세트로 하여 각 세트마다 마침표를 찍어 주어야 한다. 이 중에서 어느 하나가 빠져도 에러를 발생시킨다.

WHERE 조건에 반드시 주어-술어-목적어 패턴으로 조건을 주어야 하는데 때로는 이 패턴에서 주어가 같은 경우에는 주어를 생략할 수도 있다. 이때는 앞에 오는 패턴 끝에 ;을 사용하고 다음에 오는 패턴에는 주어를 기재하지 않고 술어-목적어만 적어 주면 된다.³⁷⁸⁾

표 90. 주어의 생략 SPARQL

<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/histor- icevent/kno#> select * where { ?event kno:agent ?Period . ?event kno:patient ?Patient . } </pre>	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/histor- icevent/kno#> select * where { ?event kno:agent ?Period ; kno:patient ?Patient . } </pre>
--	---

주어, 술어가 반복되면 목적어를 ,를 이용하여 나열할 수도 있다.

표 91. 주어·술어의 생략

<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/histor- icevent/kno#> select * where { ?event kno:agent ?Period . ?event kno:patient ?Patient . } </pre>	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/histor- icevent/kno#> select * where { ?event kno:agent ?Period, ?Patient . } </pre>
--	--

SPARQL은 LOD가 그래프 구조이기 때문에 SQL과 달리 필드를 지정하는 것이 아니라

378) 예를들면 아래와 같이 하면 된다.

```

?htimtemp rdf:type kno:Period ;
          rdf:type kno:Time .

```

변수를 지정하여 해당 변수에 바인딩 된 결과를 SELECT 뒤에 열거 해야 한다. 즉 WHERE 구문 안에 지정한 변수 중 그 결과를 바인딩하고자 하는 것을 SELECT 구문 뒤에 나열하면 된다. 전체 변수를 모두 보고 싶을 때는 '*'를 사용하면 된다. 예를 들어 '馮廷諤이 朱全忠을 시해하다', '馮廷諤이 朱友珪를 시해하다'라는 두 개의 문장을 아래와 같은 트리플로 만들 수 있다.³⁷⁹⁾

- 馮廷諤(S) 신분(P) 僕夫(O)
- 馮廷諤(S) 국적(P) 後梁(O)

이를 그래프로 표현하면 다음과 같다.

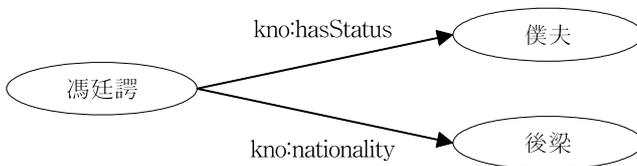


그림 88. 馮廷諤의 신분과 국적을 표현하는 RDF 그래프

이를 SPARQL에서 WHERE 뒤에 나오는 조회 조건으로 표현하면 다음과 같다. 우선 첫 번째 문장을 찾는 조회 조건은 'knoPerson:馮廷諤 kno:hasStatus ?Status .'로 표현한다. 'knoPerson, kno'는 본 논문에서 사용한 네임스페이스이며, ?Status는 임의로 정한 변수이다. 변수는 ?나 \$를 사용한다. 이들은 각각 주어, 술어, 목적어에 대응하며 맨 마지막에는 .으로 트리플이 끝났음을 표시한다. 그 다음 줄에는 두 번째 문장을 'knoPerson:馮廷諤 kno:nationality ?nationality .'로 표현한다. 여기서 두 개의 문장이 모두 주어 위치에 'knoPerson:馮廷諤'를 사용했다는 것이 중요하다. 즉 馮廷諤을 주어로 하는 RDF 트리플을 찾고, 술어는 신분(kno:hasStatus)과 국적(kno:nationality)을 찾는다. 목적어는 변수로 지정했으므로 어떤 내용인지는 모르지만 주어, 술어와 일치하는 모든 결과를 조회한다. 그리고 SELECT 뒤에 ?status ?nationality로 지정했으므로 쿼리의 결과는 status, nationality를 돌려 준다.

379) 실제 본 논문에서 설계한 구조는 이와 달리 event를 중심으로 의미역에 의해 행동주, 피동주, 행위 등을 구분하도록 되어 있다.

표 92. 馮廷諤의 신분과 국적을 표현하는 SPARQL

```

PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#>
PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent?Person/kno#>
select ?status ?nationality
where {
knoPerson:馮廷諤 kno:hasStatus ?status .
knoPerson:馮廷諤 kno:nationality ?nationality .
}
    
```

다음은 기본적인 SPARQL 명령어이다.

표 93. 기본적인 SPARQL 명령어

구분	내용
?	변수 지정할 때 ?를 사용한다.
rdf:type	트리플의 술어부에 해당하는 것으로 클래스 타입이 무엇인지를 지칭한다.
a	rdf:type과 동일한 의미한다.
.	검색 조건이 끝났음을 표시한다.
,	목적어 축약식
;	동일한 주어를 생략할 때 사용한다.
{ }	WHERE 뒤에서 사용하여 { } 안에 조건을 넣는다. { }는 Union에서 하나의 그룹을 표시한다.
Union	조회된 결과를 결합하는 명령어
DISTINCT	중복을 제거하는 명령
ORDER BY	조회된 결과를 정렬시키는 명령
LIMIT	조회된 결과를 제한할 때 사용하는 명령
OFFSET	Query 결과를 몇 번째부터 출력할 것인지 지정하는 명령

다음은 찾고자 하는 질의이다.

표 94. 유효성 검증을 위한 SPARQL 조회

연번	질의
1	만약 『通鑑節要』에서 오대십국 시대에 황제를 시해한 사람은 누가 있을까?
2	(朱友貞이 朱友珪를 쫓아내다 後梁의 황제에 올랐는데) 朱友貞과 朱友珪의 관계는 무엇인가?
3	朱全忠을 시해한 사람은 누구이며 신분은 무엇인가?
4	912년에 어떤 사건이 발생했는가?
5	馮廷諤이 어떤 행동을 했는가?

첫 번째 질의는 “『通鑑節要』에서 오대십국 시대에 황제를 시해한 사람은 누구인가?”라는 것이다. 이러한 질의에 대해서 어떻게 하면 답을 찾을 수 있을까? 이 질의에 대한 해답을 찾기 위해서는 우선 오대십국이 어떤 나라를 의미하는지 알아야 한다. 오대는 ‘後梁, 後唐,

後晉, 後漢, 後周'이며, 십국은 '南吳, 前蜀, 吳越, 南楚, 閩, 南漢, 荊南, 後蜀, 南唐, 北漢'이다. 오대십국은 이들 15개의 나라를 의미한다. 이러한 정보는 위키피디어나 각종 사전에 이미 다 나와 있는 내용이다. 이 정보면 위의 질의에 대한 답변이 가능한 정보는 모두 준비된 셈이다. 그렇다면 이들 정보를 조합하여 필요한 답을 얻을 수 있을까? 기존의 방법으로는 이들 정보를 아무리 정교하게 가공해 놓았다고 하더라도 위의 질의에 직접적인 결과를 얻기는 쉽지 않다. 이를 해결하기 위한 방법이 바로 역사적 지식을 LOD로 발행하고, 지식 관계망을 통해 검색을 확장하는 것이다.

구체적으로 질의에 대한 절차와 그 결과를 보자. 먼저 이해를 돕기 위해 배경 설명을 간단하게 하면 馮廷諤는 後梁의 郢王 朱友珪의 僕夫(마부)이면서 朱友珪의 사주를 받고 後梁 太祖 朱全忠을 시해하였다. 후에 後梁 末帝 朱友貞이 반란을 일으키자 朱友珪가 부인 張氏와 도망치다가 막다른 골목에 이르자 馮廷諤에게 자기와 부인 張氏를 죽이도록 명하자, 馮廷諤은 朱友珪를 죽이고 나서 스스로 죽는다.

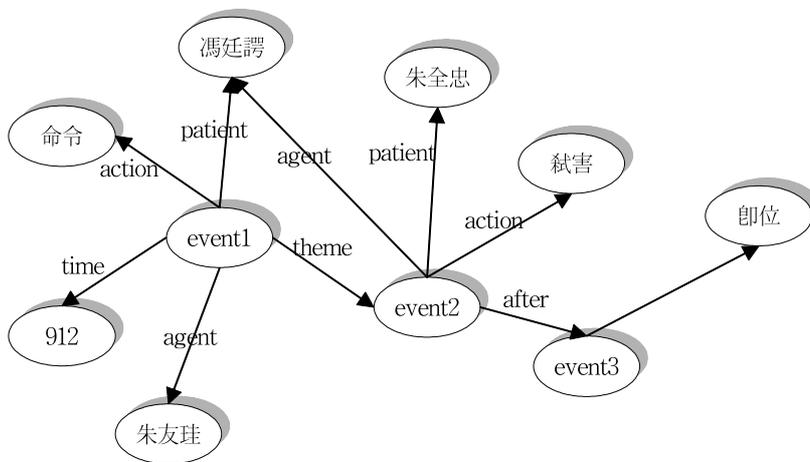


그림 89. 朱友珪가 朱全忠을 살해한 사건을 RDF 그래프 표현

먼저 朱全忠을 시해한 사람을 찾기 위한 SPARQL은 다음과 같다.

표 95. 五代十國 시대에 황제를 시해한 사람은?380)

<p>五代十國 시대에 황제를 시해한 사람</p>	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX xml: <http://www.w3.org/XML/1998/namespace> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select distinct ?dynasty ?agent ?personStatus ?person ?action ?patient ?event2 where { ?period rdf:type kno:Period . ?period kno:hasComponent ?dynasty . FILTER regex(str(?period), "五代十國") ?person kno:nationality ?dynasty . ?event2 kno:agent ?person . ?person kno:hasStatus ?personStatus . ?event2 kno:action ?action . ?event2 kno:patient ?patient . ?event1 kno:theme ?event2 . ?event1 kno:agent ?agent } </pre>
----------------------------	--

두 번째 질의는 朱友貞과 朱友珪의 관계를 묻는 것이다. 이 질의에 대한 답을 얻기 위해서는 앞서 설명한 지식 관계망을 활용해야 한다.

後梁의 太祖인 朱全忠은 자녀를 16명이나 두었다. 그 중에서 亳州營 娼妓 사이에서 둘째 아들 朱友珪를 두었으며, 正妃인 元貞皇后 사이에서 末帝 朱友貞을 두었다. 그리고 養子로 博王 朱友文이 있었다. 朱友文의 아내는 王氏로 미모가 뛰어났는데 朱全忠의 총애를 받았다고 한다. 그래서 朱全忠은 병이 심해지자 帝位를 養子인 朱友文에게 물려주려고 한다. 이에 반발한 朱友珪는 반역을 일으켜서 군사를 이끌고 朱全忠의 침실로 진입하여 僕夫人 馮廷諤를 시켜 친아버지 朱全忠을 살해하고 황제에 오른다. 이를 본 朱友貞은 얼마 후 政變을 일으켜 朱友珪를 살해하고 자신이 황제가 된다. 그런데 『通鑑節要』에서 이들의 관계를 확인하기가 쉽지 않다. 어느 곳에서는 이들 관계를 알 수 있는 내용이 있겠지만 모든 것을 그렇게 친절하게 기록해 놓지 않기 때문이다. 이들 간의 관계를 확인하기 위해서는 공구서를 참조해야 한다. 그런데 이들 자료를 찾는데 많은 시간이 소요된다. 중문 위키피디아에는 이런 황제의 가계가 다수의 집필자에 의해 잘 정리되어 있다. 그런데 이러한 노력이 좀 더 효율

380) <http://61.35.170.118:8890/sparql>

적으로 활용되기 위해서는 이를 정보를 RDF 트리플로 구축해야 한다. 일단 RDF 트리플로 구축해 놓으면 그 활용 분야가 매우 넓어지게 된다. 이러한 역할을 하는 것이 바로 지식 관계망이다. 아래와 같이 朱全忠과 관련된 가족 정보를 RDF 트리플로 구축해 놓으면, 이 정보가 필요한 모든 사람이 마치 내가 가지고 있는 DB처럼 쉽게 활용할 수 있는 것이다. 물론 가족 관계뿐만 아니라 본문 내용도 의미역을 이용하여 기술해야 한다.

표 96. RDF 트리플로 표현한 後梁 太祖 朱全忠의 가족 관계

```

<!-- http://dh.aks.ac.ke/ontology#後梁太祖朱全忠 -->

<owl:NamedIndividual rdf:about="&knoPerson;後梁太祖朱全忠">
  <rdf:type rdf:resource="&knoPerson"/>
  <kno:birthDate rdf:datatype="&xsd:string">0852129</kno:birthDate>
  <kno:deathDate rdf:datatype="&xsd:string">09120718</kno:deathDate>
  <kno:reignStart rdf:resource="&kno;907"/>
  <kno:reignEnd rdf:resource="&kno;912"/>
  <kno:birthPlace rdf:datatype="&xsd:string">宋州陽山午溝里</birthPlace>
  <kno:tombPlace rdf:datatype="&xsd:string">宣陵</tombPlace>
  <kno:hasAliasName rdf:datatype="&xsd:string">朱晃</hasAliasName>
  <kno:hasFormerName rdf:datatype="&xsd:string">朱溫</hasFormerName>
  <kno:hasPosthumousName rdf:datatype="&xsd:string">神武元聖孝皇帝
</hasPosthumousName>
  <kno:hasWife rdf:resource="&knoPerson;元貞皇后張氏"/>
  <kno:hasAdoptedSon rdf:resource="&knoPerson;冀王朱友謙"/>
  <kno:hasAdoptedSon rdf:resource="&knoPerson;博王朱友文"/>
  <kno:hasBastardDaughter rdf:resource="&knoPerson;安陽公主朱氏"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;康王朱友敬"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;建王朱友徽"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;彬王朱友裕"/>
  <kno:nationality rdf:resource="&knoDynasty;後梁"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;後梁末帝朱友貞"/>
  <kno:hasBastardDaughter rdf:resource="&knoPerson;普寧公主朱氏"/>
  <kno:hasAdoptedSon rdf:resource="&knoPerson;朱友恭"/>
  <kno:hasAdoptedSon rdf:resource="&knoPerson;朱友讓"/>
  <kno:hasBastardDaughter rdf:resource="&knoPerson;眞寧公主朱氏"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;福王朱友璋"/>
  <kno:subordinate rdf:resource="&knoPerson;蔣玄暉"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;賀王朱友雍"/>
  <kno:hasBastardSon rdf:resource="&knoPerson;郢王朱友珪"/>

```

```

<kno:hasBastardDaughter rdf:resource="&knoPerson;金華公主朱氏"/>
<kno:hasBastardDaughter rdf:resource="&knoPerson;長樂公主朱氏"/>
</owl:NamedIndividual>

```

표 97. 朱友貞과 朱友珪의 관계를 조회하는 쿼리

朱友貞과 朱友珪의 관계를 조회	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX xml: <http://www.w3.org/XML/1998/namespace> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select * where { {knoPerson:後梁郢王朱友珪 ?x knoPerson:後梁末帝朱友貞} } </pre>
------------------	---

세 번째 질의는 “朱全忠을 시해한 사람은 누구이며 신분은 무엇인가?”라는 것이다. 아래 그래프는 朱全忠이 蔣玄暉를 시켜 唐나라 昭宗을 시해한 것을 표현하고 있다.

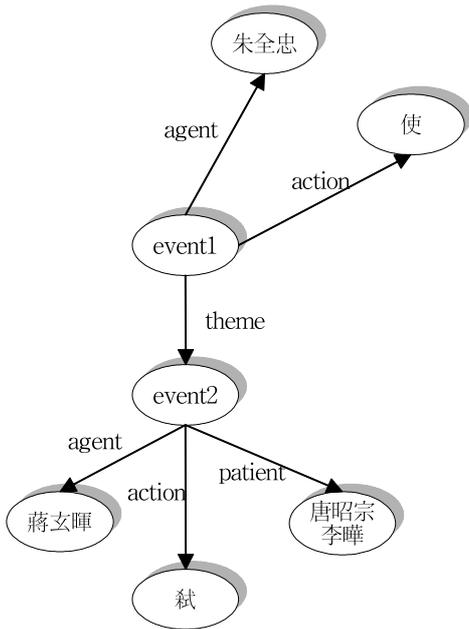


그림 90. 朱全忠이 蔣玄暉를 시켜 唐昭宗을 시해했음을 표시하는 RDF 그래프

그런데 이러한 관계의 실제 내용을 알기 위해서는 다음과 같은 쿼리가 필요하다.

표 98. 朱全忠이 蔣玄暉를 시켜 唐昭宗을 시해했음을 조회하는 쿼리

朱全忠이 蔣玄暉를 시켜 唐昭宗 李晔를 시해했음 을 찾는 쿼리	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#> PREFIX owl: <http://www.w3.org/2002/07/owl#> select ?agent ?action1 ?agent2 ?action2 ?patient where { ?event1 kno:agent ?agent . ?event1 kno:action ?action1 . ?event1 kno:theme ?event2 . ?event2 kno:agent ?agent2 . ?event2 kno:action ?action2 . ?event2 kno:patient ?patient . FILTER regex(str(?action2), "弑"). FILTER regex(str(?patient), "昭宗") } </pre>
---	---

네 번째 질의는 “912년에 중국에서 어떤 사건이 발생했는가?”라는 것이다. 이에 대한 적절한 답변을 찾기 위해서는 조회 조건을 사건이 발생한 시간(time)과 행위자와 행위, 그리고 행위의 영향을 받는 사람으로 좁혀야 한다. 우선 행위나 상태와 관련된 본문 내용은 이벤트를 중심으로 목적어를 연결하였으므로 주어 위치에는 항상 event가 위치해야 한다. 그래서 주어 위치에 ‘?event’로 임시 변수를 지정하였다. 술어 위치에는 찾고자 하는 의미역이 와야 한다. 여기서는 행위를 한 자, 행위가 발생한 시간, 행위, 행위를 받은 피동주를 각각 ObjectProperty ‘kno:agent, kno:time, kno:action, kno:patient’로 정확하게 기술해 주어야 한다. 목적어에는 오브젝트 프로퍼티가 연결되어 있는 대상을 불러올 수 있도록 변수를 지정한다. 그런데 모든 사건을 가져 오는 것이 아니라 912년에 발생한 사건을 조회하기 위해서 시간을 ‘?event kno:time kno:912.’처럼 구체적으로 명시하였다. 그 결과 ‘馮廷諤’이 ‘朱全忠’을 시해하였음을 확인할 수 있다.

표 99. 912년에 발생한 사건을 조회하는 쿼리

<p>912년에 어떤 사건이 발생했는가?</p>	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX xml: <http://www.w3.org/XML/1998/namespace> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select ?agent ?action ?patient where { ?event kno:agent ?agent. ?event kno:time kno:912. ?event kno:action ?action. ?event kno:patient ?patient } </pre>
----------------------------	---

마지막 질의는 “馮廷諤이 어떤 행위를 했는가?”이다. 위의 첫 번째 질의로부터 event1의 행동주인 朱友珪가 피동주 馮廷諤에게 명하여 朱全忠을 시해했다는 사실을 알 수 있다. 그렇다면 馮廷諤이 또 어떤 행위를 했는지 알아보자. 온톨로지 설계에서 모든 행위는 event를 주어로 하여 각각이 의미역으로 연결이 된다고 하였다. 이 때문에 주어 위치에는 Event 인스턴스가 오며, 행동주와 행위, 피동주를 각각 변수로 지정하여 그 결과를 바인딩하였다. 이들은 개별적인 event로 구성되었으며, event1의 대상은 event2가 된다. 즉 朱友珪가 시킨 그 내용이 바로 event2가 된다.

표 100. 馮廷諤이 어떤 行爲를 했는지 찾는 쿼리

<p>馮廷諤이 누구에게 어떤 행위를 했는지 찾는 쿼리</p>	<pre> PREFIX kno: <http://dh.aks.ac.kr/ontologies/historicevent/kno#> PREFIX knoPerson: <http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX xml: <http://www.w3.org/XML/1998/namespace> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> select * where { ?event kno:agent ?agent . ?event kno:action ?action . ?event kno:patient ?patient . FILTER regex(str(?agent), '馮廷諤') } </pre>
-----------------------------------	--

위의 쿼리를 실행시킨 결과 아래 그림과 같이 馮廷諤이 朱全忠과 朱友珪을 시해했음을 확인할 수 있다.

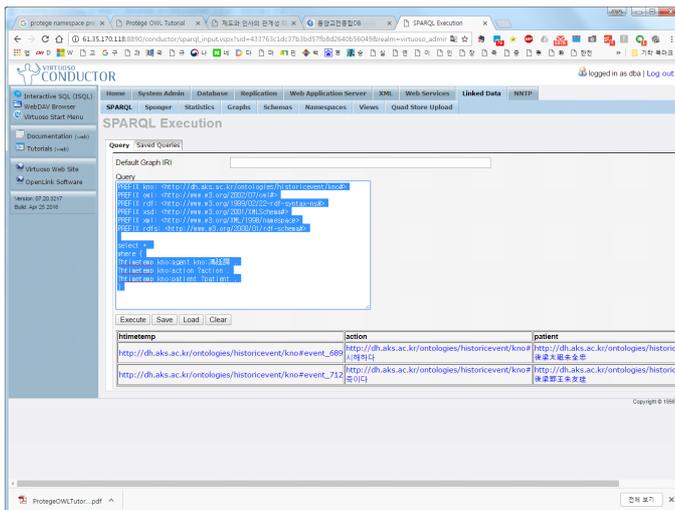


그림 91. 馮廷諤이 행위를 했는지 찾는 쿼리

V. 결론

정보화 사회에서 데이터는 ‘새로운 오일(data is the new oil)’로 불린다. 이는 데이터의 경제적 가치에 주목한 결과이다. 이때의 경제적 가치는 데이터 간의 연계와 무한 확장, 데이터 활용 주체 간의 협력, 다양한 조합·분석을 통해 발생하며, 이를 구체화해줄 기술이 바로 LOD이다. 그러나 지금까지 진행된 LOD 서비스는 대부분 도서관의 서지 정보나 메타데이터와 같은 RDB 형태의 정형화된 데이터를 중심으로 이루어졌다. 그 원인을 살펴보면 아직 LOD로 발행하는 사업이 막 시작된 초기 단계이기에 LOD의 발행 대상이 단기간에 성과를 낼 수 있는 정형화된 데이터에 집중되었기 때문일 것이다. 현재의 LOD 발행은 일부를 제외하고는 대부분 정부 주도로 진행되고 있다. 정부의 제5차 국가정보화 기본계획(2012~2017)³⁸¹⁾에 LOD 추진 계획이 포함되면서, 2013년도에 생물, 한국사 분야에 처음으로 LOD 시범 사업이 시작되었다. 이때 사업 추진 목적을 산업적 활용도가 높고, 신규 비즈니스 창출이 가능한 분야를 발굴하는 것으로 했기 때문에 아무래도 그 영역이 제한적일 수밖에 없었다. 그리고 인문학 분야에서 LOD를 이용하여 의미 있는 성과를 내놓을 수 있을 정도의 연구 역량이 충분히 축적되지 않은 것도 하나의 이유가 될 수 있을 것이다.

LOD 발행은 어느 한 개인의 노력으로 완성될 수 있는 것이 아니다. LOD는 특정 기관이나 조직의 소유물은 아니지만, 관심사를 공유하는 조직적인 노력 없이는 만들어질 수 없는 것이다. 다른 기관이나 개인과 협업을 통해서 새로운 LOD를 만들어 낼 수도 있다. 이는 LOD가 가진 가장 큰 장점이기도 하다. 그렇기 때문에 앞으로 LOD를 적용하는 영역도 확대되고 그 깊이도 깊어질 것으로 생각한다. 그러면 역사 분야도 자연스럽게 메타데이터를 LOD로 전환하는 수준에서 점차 사료에 나와 있는 내용을 기술하는 것으로 관심이 옮겨 갈 것이다. 그런데 역사적 지식을 LOD로 발행하는 데는 일반적인 RDB를 RDF 트리플로 변환하는 것에 비해 훨씬 더 많은 노력과 시간이 필요하다. 역사적 사건은 기본적인 어휘 간의 관계뿐만 아니라 시간의 선후 관계나 인과 관계, 관련 공간 등 거의 모든 정보를 정밀하게 확인하고 표시해 주어야 하기 때문이다.

사료를 직접 기술할 수 있는 LOD를 만들기 위해서는 무엇보다도 정리된 온톨로지 모델이 있어야 한다. 온톨로지 모델은 결국 시맨틱 규칙을 적용할 수 있는 노드와 릴레이션을 찾아내는 것이 핵심이며, 이 모델이 의미가 있음을 확인하기 위해서는 많은 데이터를 대상으로 하여 실제로 적용해 보고 그 문제점을 보완해야 한다. 이론적으로 아무리 뛰어나더라도 실제로 구현에 들어가면 데이터와 일치하지 않는 경우가 많기 때문이다. 본 연구는 이러한 역사적 지식을 LOD로 표현할 수 있도록 방법을 제시하고 있으며, 이를 위해서 다음과

381) 미래창조과학부, 「제5차 국가정보화 기본 계획(2013~2017)」, 2013, 35쪽.

같은 연구를 수행하였다.

첫째, 역사 자료를 연계 및 통합하기 위한 방법으로 행위와 상태를 중심으로 하여 LOD로 발행할 수 있는 온톨로지 모델을 제시하였다. 역사적 지식을 LOD로 발행하는 이유는 사람뿐만 아니라 기계가 이해할 수 있는 데이터 중심의 웹을 만들기 위해서이다. LOD는 RDF 트리플을 기반으로 하고 있다. RDF는 주어-술어-목적어가 하나의 의미를 구성하는 데이터 셋이다. 그러나 주어-술어-목적어의 단순한 二項 관계로는 복잡한 역사적 지식을 표현하는 데 많은 제한이 있을 수밖에 없다. 이를 해결하기 위한 방안으로 본 논문에서는 국어 문법에서 사용하는 의미역(semantic role)이란 개념을 도입하였다. 意味役이란, 어떤 동사나 형용사가 하나의 문장을 완성하기 위해서 필수적으로 요구하는 성분을 필수 보어(complements)라고 하는데, 이 필수 보어와 술어 사이의 관계를 말한다. 意味役의 도입은 하나의 사건을 서술어와 나머지 보어와의 관계로 표현함으로써 기계적으로 추론이 가능한 RDF를 만들 수 있게 하였다. 그리고 意味役의 적용은 기존 국어학에서 문장을 문법적으로 분석하는 것과는 달리, 엄밀한 문법의 적용보다 정확한 내용의 전달에 중점을 두었다.

둘째, LOD 발행으로 해결하기 어려운, 지능화된 연계를 위해서 지식 관계망(Knowledge Network)을 제안하고 그 온톨로지 모델을 제시하였다. 지식 관계망은 하나의 지식 노드와 관련된 다른 노드를 서로 연결해서 분절화된 지식을 의미적으로 연결해 주는 네트워크를 의미한다. 이러한 지식 관계망은 지식의 개별 노드와 노드를 연결해서 새로운 지식으로 이동할 수 있는 소통의 교점을 마련해 준다. 지식 관계망을 구축하기 위해서는 지식의 寶庫인 辭典 등에서 수집한 어휘 간의 관계, 同義語, 反意語 등의 의미론적 연관 관계와 역사의 주체인 인물 간의 관계 등을 정밀하게 정제하고, RDF로 표현해야 한다. 그 결과 지식 관계망을 통해서 기존 LOD로 해결할 수 없는 훨씬 더 다양하고 복잡한 지식으로 접근이 가능하게 될 것이다.

셋째, 역사적 지식을 행위와 상태를 중심으로 하여 LOD로 표현하기 위해 『通鑑節要』 중의 五代十國 시대를 대상으로³⁸²⁾ LOD로 발행해 보았다. 『通鑑節要』에 나오는 역사적 지식을 LOD로 발행하는 구체적인 방법론과 절차를 세부적으로 제시하였으며, 실제로 데이터를 어떻게 가공하고, 어떤 점을 고려해야 하는지를 정밀하게 검토하였다. 구체적인 방법으로 먼저 역사서에서 사용되는 개체명을 파악하기 위해 『通鑑節要』를 대상으로 본문에 개체명을 마크업 하였으며, 이를 통해 역사서에 사용된 개체명의 유형을 파악하고 클래스 설계에 반영하였다. 정확한 온톨로지 설계를 위해 설계와 적용, 보완 등의 순환 반복 과정을 걸쳐서 온톨로지 모델을 정교하게 가다듬었다. 역사적 지식을 意味役으로 표현하기 위한 방법으

382) 意味役을 이용한 역사 지식을 LOD로 구축하는 것은 『通鑑節要』 전체 50권 중에서 48권, 49권, 50권 등 모두 3책을 대상으로 하였다.

로는, 작업의 효율성을 위해 XML 문서 내에 직접 意味役을 속성으로 마크업을 하는 방법을 적용하였다. 그리고 역사적 지식을 LOD로 발행하여 SPARQL을 이용해서 역사적 지식을 표현하는 데 유용함을 보여주었다.

또한 중간 과정의 산출물을 모든 사람이 참조할 수 있도록 위키사이트³⁸³⁾에 제공함으로써 본 연구가 단순히 연구에만 그치지 않고, 관심 있는 연구자들이 실제로 적용하고 확대 발전시킬 수 있게 하여 이 분야의 연구에 조금이라도 도움을 주고자 하였다. 그리하여 누구든지 중간 과정의 산출물을 다운받아서 필요한 정보를 추가할 수도 있고, 수정할 수도 있다. 이러한 과정에서 나타난 한계도 많이 있었다.

넷째, 동아시아 한자문화권 국가에서 서로 간에 지식의 소통이 원활하게 이루어질 수 있도록 하는 가능성을 모색하였다. 한국, 일본, 중국 등의 국가들은 근대 이전에 한문을 이용해 역사를 기록했다는 특징이 있다. 그러면서 이들 자료를 학습하거나 활용할 경우에는 번역문을 이용하였다. 그렇기 때문에 역사 사료에서 한문 원전에 쓰여진 표현과 그 의미를 적절하게 현대적으로 해석했을 때 쓰는 표현 간의 상관 관계에 대해서도 우리가 지속적으로 연구할 필요가 있다. 구체적으로 event 노드를 비롯해 한문 원전에 사용된 어휘를 번역문과 연결시키는 작업을 진행하였다. 한문 원문에 대한 번역은 각국의 언어별, 시대별로 새롭게 만들어 지고 있으며, 번역하는 사람에 따라 그 표현이 달라지기 마련이다. 이런 번역에 사용된 상이한 표현들을 지속적으로 LOD로 축적하면 향후 자동 번역에도 큰 도움이 될 것이다. 이때 중요한 것은 어휘의 URI 부여이다. 본 연구에서는 원문에 사용된 한자짜리 한자는 『漢韓大辭典』의 의미 구분을 따르도록 하였다.

본 연구를 하면서 나타난 한계를 요약하면 다음과 같다. 첫째, 적용한 LOD로 발행한 대상 시대가 짧음을 들 수 있다. LOD로 발행한 대상이 『通鑑節要』 중 唐 宣宗이 즉위한 846년부터 宋 태조가 즉위하기 직전인 960년까지 약 1백 년이 조금 넘는 기간이다. 책의 분량으로 보면 전체 50권으로 구성된 『通鑑節要』 중에 3권에 해당한다. 그러나 역사를 구성하고 있는 지식이 복잡하더라도 서술어를 중심으로 문장을 분해하면 아주 짧은 단위로 구분할 수 있다. 이 구분한 단위에서 서술어가 필요로 하는 補語가 그렇게 많은 경우의 수가 나올 수는 없다. 그렇기 때문에 본 논문에서 제시하는 意味役을 통해서 대부분의 행위를 표현할 수 있을 것이라고 생각되지만 아무래도 짧은 시대를 다루었고, 그 분량이 많지 않기 때문에 본 논문에서 수용하지 못하는 경우가 발견될 수도 있을 것이다.

둘째, 五代十國이라는 시대의 특수성을 들 수 있다. 五代十國이라는 시기는 戰亂이 빈번하고 정치적 변화가 극심했던 시기였다. 血肉 간의 骨肉相殘이 빈번하게 발생하였으며, 황제를 弑害하고 새로운 왕조를 개창한 것도 흔한 일이었다. 왕조의 교체가 얼마나 심한지 劉

383) <http://dhlabs.aks.ac.kr/~yju2999/wiki/index.php>

知遠이 개국한 後漢(947~950년)은 불과 3년 만에 망하였으며 이는 중국 역사에서도 가장 단명한 왕조이다. 이러한 시기였기 때문에 이벤트를 나타내는 행위 동사가 ‘戰爭, 戰鬥, 侵略, 殺人’ 등과 ‘官職의 임명, 해임’ 등에 관련된 어휘가 특히 많았으며 상대적으로 사건을 표현하기도 용이했던 측면이 있다. 그러나 행위가 역사적으로 중요한 사건임은 틀림없지만 역사서에는 이런 행위 이외에도 다양한 내용이 포함되어 있다. 예를 들어 朝廷 안에서의 여러 議論, 上疏文, 制度의 制定이나 施行 등도 역사를 구성하는 중요한 요소이다. 이러한 역사적 사건은 상대에서 물리적인 충격이 가해지는 행위는 아니지만, 역사적으로 일정한 영향을 발생시키고, 사회적으로 변화를 가져오기도 한다. 그렇기 때문에 이러한 내용까지 본 모델에 의해서 표현할 수 있음을 보여주었어야 하는데 『通鑑節要』가 담고 있는 내용으로는 이 부분까지 실증하기에는 충분하지 않을 것으로 판단된다.

마지막으로 역사서에 나오는 행위를 Action 클래스의 인스턴스에 의해 표현한다는 것은 동사를 일정한 개념 체계로 설정해 주어야 그 활용도가 높아진다는 것을 의미한다. 국내에서는 ‘코어넷’이나 ‘세종전자사전’이 의미역 분류를 적용하였으나 이들의 의미역 분류는 한자어를 많이 포함하고 있는 역사서에 적용하기에는 그 분류 체계가 단순하다. 그렇기 때문에 한자어를 포함한 사건을 나타내는 동사 위주로 별도의 분류 체계를 만들어 주었어야 하는데, 그 부분이 하나의 큰 연구 영역이기 때문에 아쉽지만 다음 연구로 미룰 수 밖에 없었다.

본 연구는 역사 지식을 LOD로 표현하는 데 많은 과제를 남겨두었지만, 『通鑑節要』라는 역사서를 대상으로 역사 지식을 LOD로 발행함으로써, 부족하지만 역사 지식의 LOD로 발행하는 첫걸음을 디뎠다는데 그 의의가 있다고 할 수 있다.

VI. 참고문헌

1. 자료

- 李賢, 『大明一統志』
司馬光, 『資治通鑑』
朱熹, 『資治通鑑綱目』
司馬遷, 『史記』
江贇, 『通鑑節要』
薛居正, 『舊五代史』

2. 단행본

- David Wood·Marsha Zaidman·Luke Ruth(저)/오원석·이명진·박진호·김보람(역), 『Linked Data - Structured data on the web』, 2014.
- Dean Allemang·James A. Hendler(저)/김성혁·박영택·추윤미(역), 『온톨로지 개발자를 위한 시맨틱 웹』, 사이텍미디어, 2008.
- Vossen Piek(저)/한정환·최경봉·도원영·이봉원·이동혁·이현희·김혜영(역), 『유로워드넷』, 한국문화사, 2004.
- 고영만·김비연·민혜령·송민선·이승준, 『구조적 학술용어사전 기반 온톨로지 구축론』, 한국도서관협회, 2016.
- 곽기영, 『소셜네트워크분석』, 청람, 2014.
- 溝口理一郎(저)/최기선·황도삼(역), 『차세대 웹과 지식 처리의 핵심 기술』 온톨로지 공학』, 두양사, 2009.
- 김현, 『인문정보학의 모색』, 북코리아, 2012.
- 김현·임영상·김바로, 『디지털 인문학 입문』, 한국외국어대학교 지식출판원, 2016.
- 남승호, 『한국어 술어의 사건 구조와 논항 구조』, 서울대학교출판부, 2007.
- 노상규·박진수, 『인터넷 진화의 열쇠 온톨로지』, 가즈토이, 2009.
- 마쓰오 유타카(저)/박기원(역), 『인공지능과 딥러닝』, 동아엠앤비, 2015, 102쪽.
- 박종호, 『한국어 동사 의미망 구축 방법론』, 청운, 2012.
- 백옥인, 『디지털 데이터·정보·지식』, 커뮤니케이션북스, 2013.
- 司馬光, 『資治通鑑』, 中華書局, 1956.
- 成百曉, 『譯註 通鑑節要』, 전통문화연구회, 2006

- 神崎正英(저)/황성역·양해술(역), 『시맨틱 웹을 위한 RDF/OWL 입문』, 홍릉과학출판사, 2008.
- 王力(저)/李鴻鎮(역), 『中國古代文化常識』, 螢雪出版社, 1994.
- 劉節(저)/辛太甲(역), 『中國史學史』, 서신원, 2000.
- 伊藤健太郎·佐藤勇紀·濱崎俊(저)/정재은·이인근·황도삼(역), 『시맨틱 웹』, 두양사, 2015.
- 이성무 외, 『조선을 이끈 명문가 지도』, 글항아리, 2011.
- 이숙의, 『한국어 동사 온톨로지 구축 연구』, 역락, 2013.
- 이종동·조성을, 『중국사학사』, 혜안, 2009.
- 齋藤孝(저)/최석두·한상길(역), 『온톨로지 알고리즘』1, 한울아카데미, 2008.
- 朱熹(저)/신승운(역), 『(譯註) 思政殿訓義 資治通鑑綱目』, 전통문화연구회, 2011.
- 中砂明德(저)/강길중·김지영·장원철(역), 『우아함의 탄생』, 민음사, 2009.

3. 학위논문

- 강민아, 「NTIS 연구정보 개방을 위한 LOD 구축 및 활용 방법 구현」, 충남대학교 대학원 석사학위논문, 2015.
- 강현민, 「RDF/OWL의 객체속성을 이용한 관계온톨로지 시스템 구축과 활용에 관한 연구」, 연세대학교 대학원 석사학위논문, 2010.
- 김상균, 「TL-OWL : 시간 지식의 표현과 추론을 지원하는 온톨로지 언어」, 충남대학교 대학원 박사학위논문, 2008.
- 김우임, 「서울·경지지역의 朝鮮時代 士大夫 墓制 研究」, 고려대학교 대학원 박사학위논문, 2007.
- 나방현, 「U-City 정보서비스의 접근성 향상 방법에 관한 연구」, 한세대학교 대학원 박사학위논문, 2010.
- 박진호, 「도서관 데이터의 링크드 데이터(Linked Data) 변환과 인터링킹(interlinking)을 통한 정보연계 확장성에 관한 연구」, 성균관대학교 일반대학원, 석사학위논문, 2013.
- 박현규, 「대용량 온톨로지를 위한 시간 정보 표현 및 추론」, 숭실대학교 대학원 석사학위논문, 2014.
- 서소리, 「문화유산 지식 정보 데이터 모델 연구」, 한국학중앙연구원 한국학대학원 석사학위논문, 2014.
- 이숙의, 「한국어 동사 의미망 구축 연구」, 충남대학교 박사학위논문, 2006.

- 이현정, 「링크드 데이터 활용을 위한 멀티소스 검색시스템 구축에 관한 연구」, 중앙대학교 대학원 석사학위논문, 2015.
- 최주원, 「문서 기반 온톨로지 생성을 위한 문장 분해 및 유사도 측정 시스템」, KAI ST 석사학위논문, 2009.
- 한상은, 「오픈데이터의 링크드데이터(Linked Data) 발행방법 간 유용성 비교평가에 관한 연구」, 성균관대학교 석사학위논문, 2014.
- 황인혁, 「조선시대 『濬源寶鑑』의 분석과 활용에 관한 연구」, 건국대학교 대학원 박사학위논문, 2015.

4. 연구논문

- 「국가과학기술 지식정보 유통서비스를 위한 시맨틱 기술」, 『지식정보인프라』 26호, 2007, 23~31쪽.
- Linked Data를 중심으로」, 『한국정보관리학회 학술대회논문집』, 2014, 109~114쪽.
- 강대현, 이정민, 김태윤, 박한샘, 권경락, 정인정, 「메타데이터를 사용한 역사 온톨로지 생성 방법에 관한 연구」, 2015, 801~820쪽.
- 강민서·김재성·김선동·이재길, 「한글 DBpedia 온톨로지 스키마 구축」, 『한국정보과학회언어공학연구회 2014년도 제26회 한글 및 한국어 정보처리 학술대회』, 2014, 139~142쪽.
- 강신재, 「데이터의 웹을 위한 상호연결된 대규모 온톨로지 네트워크 구축」, 『한국산업정보학회논문지』, 제15권, 제1호, 2010, 15~23쪽.
- 강인수·정한민·이승우·김평·이미경·성원경, 「시맨틱 웹 온톨로지에서의 OWL sameAs 적용」, 『정보과학회논문지 : 소프트웨어 및 응용』, 제34권, 제4호, 2007, 359~367쪽.
- 강현민, 「공리정의를 이용한 인명접근점제어 온톨로지 기술에 관한 연구」, 『한국문헌정보학회지』, 제46권, 제2호, 2012, 157~174쪽.
- 곽정애·용환승, 「워드넷의 의미 관계 집합을 이용한 온톨로지 매핑」, 『정보과학회논문지 : 데이터베이스』, 제36권, 제6호, 2009, 466~475쪽.
- 권중달, 「『자치통감』의 사학사적 의미」, 『한국사학사학보』 21호, 2010, 205~249쪽.
- 권창희, 「U-City에 있어서 U-Service의 콘텍스트 기반 역사문화 온톨로지 설계 관한 연구」, 『디지털정책연구』, 제9권, 제6호, 2011, 379~387쪽.
- 김수경, 안기홍, 최호진, 「새로운 N-ary 관계 디자인 기반의 온톨로지 모델을 이용한 문장의미결정」, 『정보관리학회지』, 제25권, 제4호, 2008, 43~66쪽.
- 김윤정·옥철영, 『한국어 서술어와 논항들 사이의 의미역』, 2014, 143~148쪽.

- 김윤조, 「조선 후기 지식인들의 『通鑑節要』에 대한 비판적 인식의 양상과 의미」, 『한문학보』, 제5권, 제0호, 2001, 239~264쪽.
- 김지영·박선아·이선희, 「과학인물 LOD 구축에 관한 연구」, 『한국도서관·정보학회지』, 제45권, 제4호, 2014, 429~455쪽.
- 김현, 「한국 고전적 전산화의 발전 방향」, 『민족문화』, 제28집, 한국고전번역원, 2005, 157~186쪽.
- 류철중·윤보현·김용, 「언어자원 자동 구축을 위한 위키피디아 콘텐츠 활용 방안 연구」, 『디지털융복합연구』, 제13권, 5호, 187~194쪽.
- 박원주, 남궁현, 조기성, 류원, 「링크드 데이터 기술 기반의 공공 오픈 데이터 기술 동향」, 『주간기술동향』, 정보통신산업진흥원, 2013, 1~135쪽.
- 박지영, 「온톨로지 개발을 위한 일반 지침(Ontology Development 101: A Guide to Creating Your First Ontology)」, 『도서관 정책자료』, 2008, 1~37쪽.
- 박철우·김중명, 「한국어 용언 사전 기술을 위한 의미역 설정의 기본 문제들」, 『語學研究』, 제41권, 제3호, 2005, 543~567쪽.
- 배영준·옥철영, 「한국어 어휘지도(UWordMap)와 API 소개」, 『한국정보과학회언어공학연구회 2014년도 제26회 한글 및 한국어 정보처리 학술대회』, 2014년, 27~31쪽.
- 오삼균, 「온톨로지 언어의 비교 연구 : W3C OWL과 ISO 토픽맵을 중심으로」, 『한국비블리아학회지』, 제15권, 제2호, 2004, 71~96쪽.
- 이계명, 「자치통감」, 『역사학연구(구 전남사학)』, 제12권, 1998, 119~168쪽.
- 이인근·정재은·황도삼, 「Y-HisOnto: Q&A 시스템에서의 활용을 위한 역사 온톨로지 모형」, 『한국정보과학회언어공학연구회 2013년도 제25회 한글 및 한국어 정보처리 학술대회』, 제25권, 제1호, 2013, 156~159쪽.
- 이현정·남영준, 「국내 Linked Open Data 발행 후 활용에 관한 연구: 국가서지와 RiSS」
- 이현주, 「학술연구 : 국립중앙도서관 국가서지 LOD 구축 사례」, 『디지털도서관』, 제77권, 2015, 21~35쪽.
- 이혜원·윤소영, 「역사용어 온톨로지 모형 적용 방안 연구」, 『한국비블리아학회지』, 제22권, 제1호, 2011, 263~280쪽.
- 임경태·함영균·Rezk Martin·박정열·윤용운·최기선, 「데이터 웹을 위한 한국어 DBpedia리소스 접근방법 연구」, 『한국정보과학회 2012 가을 학술발표논문집』, 제39권, 제2호(B), 2012, 112~114쪽.
- 임규홍, 「국어 정도 부사의 화용화」, 『언어과학연구』, 제24집, 2003, 283~302쪽.
- 정도현·황명권·조민희·정한민·윤소영·김경선·김평, 「온톨로지와 텍스트 마이닝 기반

- 지능형 역사인물 검색 서비스, 『인터넷정보학회논문지』, 제13권, 제5호, 2012, 33~43쪽.
- 정승화, 「인터넷 백과사전 위키피디아와 지식권력의 재구성」, 『人文科學』, 106권, 2016, 87~119쪽.
- 정한민·강인수·구희관·이승우·성원경, 「URI 서버에 기반한 국가 R&D 기반정보 온톨로지 설계 및 구현」, 『정보관리연구』, 제37권, 제2호, 2006, 109~136쪽.
- 조명대, 「조선왕조실록 Linked Data 발행에 관한 연구」, 2012, 123~138쪽.
- 조원희, 「디지털 인문학과 元代史(원대사) 研究(연구) -中國傳記資料庫(CBDB)의 소개와 전망을 중심으로」, 『중앙아시아연구』, 제21권, 제1호, 2016, 117~138쪽.
- 최지예·김명근·박소영, 「문화유산정보 말뭉치 구축을 위한 개체명 및 이벤트 부착 도구」, 『韓國컴퓨터情報學會論文誌』, 2012, 29~38쪽.
- 한미경, 「웹 기반 중국인물정보원의 현황 조사와 분석 연구」, 『한국문헌정보학회지』, 제46권, 제4호, 2012, 243~263쪽.
- 허태용, 「17세기 후반 正統論의 강화와 『資治通鑑節要』의 보급」, 『한국사학사학보』, 제2권, 2000, 5~40쪽.
- 황미녕, 정도현, 최성필, 조민희, 정한민, 「SKOS를 이용한 시소러스의 온톨로지 모델링과 LOD 공개」, 『한국정보과학회 2012한국컴퓨터종합학술대회 논문집』, 제39권, 제1호, 2012, 92~94쪽.
- Allen, James F, George Ferguson, 「Actions and events in interval temporal logic」, 『Journal of logic and computation』, 4 5, 1994.
- Gruber, T.R. 1993. 「A translation approach to portable ontologies」, Knowledge Acquisition, 5(2).
- Lyons, J. 『Semantics』, Cambridge University Press, 1977.
- Radinsky, Kira, Sagie Davidovich, Shaul Markovitch, 「Learning to predict from textual data」, 『Journal of Artificial Intelligence Research』, 45, 2012.
- Vossen, Piek, 「Version 3 Final July 1, 2002 Piek Vossen (ed.) University of Amsterdam」, 2002.
- Wang, Yandong, Jingjingf Dai, Jizhen Sheng, Kai Zhou, Jianya Gong, 「Geo-ontology design and its logic reasoning」, 『Geoinformatic 2007: Geospatial Information Science』, 2007.

5. 보고서

공간정보연구원, 대한지적공사, 「역사문화지도 구현을 위한 PROTOTYPE 개발 (최종보고서)」, 2014.

국립국어원, 「21세기 세종계획 전자사전 개발」, 2007.

국토지리정보원, 「지명 표준화 편람」, 국토지리정보원, 2012.

김현, 「국립한글박물관 디지털 아카이브 구축 기본 구상」, 한국학중앙연구원 한국학대학원 문화예술학부, 2013.

미래창조과학부, 「제5차 국가정보화 기본 계획(2013~2017)」, 2013.

온톨로지연구소, 「국가지식정보 온톨로지 표준개발」, 2006.

한국교육학술정보원, 「KEM 고도화를 위한 온톨로지 기반 시맨틱 웹 연구」, 2006.

한국데이터베이스진흥센터, 「디지털 콘텐츠 식별자 구문구조 연구」, 정보통신부, 2000.

한국문화정보센터, 「지능형 웹 기반 문화정보 활용 활성화 연구 보고서」, 2011.

한국전산원, 「웹 온톨로지 개발지침 연구」, 2004.

한국정보통신기술협회, 「OWL 웹 온톨로지 언어 참고서」, 2008.

한국정보통신기술협회, 「개체명 태그세트 및 태깅 말뭉치(Tag Set and Tagged Corpus for Named Entity Recognition)」, 2015.

한국정보화진흥원, 「2014 링크드 오픈 데이터 국내 구축 사례집」, 2014.

한국정보화진흥원, 「LOD 기반의 데이터 관리 패러다임 전환 전략」, IT & Future Strategy(1호) 2014. 4.

한국정보화진흥원, 「LOD 기반의 데이터 관리 패러다임 전환 전략」, 『IT & Future Strategy 보고서』, 1, 2014.

한국정보화진흥원, 「데이터베이스 활용기술전망 부록 :데이터 활용 신기술 가이드 1: RDF」, 2014.

한국정보화진흥원, 「링크드 데이터 구축 공정 가이드 v1.0」, 2014.

한국정보화진흥원, 「알기 쉬운 Linked Open Data」, 2015.

ICOM/CIDOC Documentation Standards Group, 2001

6. 발표자료

김현·윤종웅·안성수, 「동양고전어휘정보화의 방법 및 활용 방안」, 전통문화연구회, 2013.

오원석, 「Linked Data 4 Principles」, (주)탐쿼드란트코리아, 2015.

이경옥, 「LOD InterLinking」, 다이퀘스트, 2015.
이경일, 「온톨로지 구축과 의미 메타데이터 관리」, 솔트룩스, 2006.
이명진, 「링크드 오픈 데이터 -소개부터 구축 및 활용까지」.

7. 웹 사이트

RISS linked open data, 한국교육학술정보원, <http://data.riss.kr>
고전번역서, 한국고전번역원, <http://db.itkc.or.kr/dir/item?itemId=BT#/dir/list?itemId=BT&gubun=book>
낱말창고, (주)낱말 어휘정보처리연구소, <http://www.wordnet.co.kr>
동양고전번역지원, 전통문화연구회, <http://lu.juntong.or.kr/juntong>
동양고전종합DB, 전통문화연구회, <http://db.cyberseodang.or.kr>
維基百科, <https://zh.wikipedia.org>
인문정보학 온톨로지 설계 가이드라인, http://kadhlab102.com/wiki/index.php/인문정보학_온톨로지_설계_가이드라인
주소데이터LOD, 주식회사리스트, <http://jusodata.kr>
中國歷代人物傳記資料庫, 하버드대학교, <http://projects.iq.harvard.edu/chinesecbdb/home>
코아넷, KAIST, http://semanticweb.kaist.ac.kr/org/bora/CoreNet_Project/index.html
표준국어대사전, 국립국어원, <http://stdweb2.korean.go.kr/main.jsp>
한국 더블린코아 DCMI 공인사이트, <http://www.dublincore.go.kr/metadata-basics/term>
한국고전종합DB, 한국고전번역원, <http://db.itkc.or.kr>
한국근현대인물자료, 국사편찬위원회, <http://db.history.go.kr/item/level.do?itemId=im>
한국문집총간, 한국고전번역원, <http://db.itkc.or.kr/dir/item?itemId=MO#/dir/list?itemId=MO&gubun=book>
한국민족문화대백과사전, 한국학중앙연구원, <http://encykorea.aks.ac.kr>
한국사 LOD, 국사편찬위원회, <http://lod.koreanhistory.or.kr/lodIntro.do>
한국역사용어시소러스, 국사편찬위원회, <http://thesaurus.history.go.kr>
한국역사정보통합시스템, 국사편찬위원회, <http://www.koreanhistory.or.kr>
한국향토문화전자대전, 한국학중앙연구원, <http://www.grandculture.net>

「The top 500 sites on the web」, Alexa, <http://www.alexa.com/topsites>
 Accessing Relational Databases as Virtual RDF Graphs, <http://d2rq.org>
 Altova XMLSpy 홈페이지, <https://www.altova.com>
 baidu, <https://www.baidu.com>
 China Historical GIS, <http://www.fas.harvard.edu/~chgis>
 China Historical GIS, 하버드대학교, <http://www.fas.harvard.edu/~chgis/>
 CIDOC CRM Special Interest Group., <http://www.cidoc-crm.org/>
 CIDOC CRM, ICOM/CIDOC, <http://www.cidoc-crm.org/Version/version-6.2>
 CIDOC CRM, International Council of Museums, <http://www.cidoc-crm.org>
 Dan Brickley, Libby Miller, 「FOAF Vocabulary Specification 0.99」, 2014, <http://xmmlns.com/foaf/spec>
 DBpedia, <http://wiki.dbpedia.org>
 emeditor, Emurasoft, <https://ko.emeditor.com>
 Franz Inc., <http://franz.com/agraph/gruff>
 Getty Thesaurus of Geographic Names® Online, <http://www.getty.edu/research/tools/vocabularies/tgn>
 LIMES, AKSW, <http://aksw.org/Projects/LIMES.html>
 Neo Technology Inc., <https://neo4j.com>
 OPENLINK SOFTWARE, <http://linkeddata.uriburner.com/fct>
 OWL Manchester, OWL Validator, <http://mowl-power.cs.man.ac.uk:8080/validator>
 OWL Syntax Converter, <http://mowl-power.cs.man.ac.uk:8080/converter>
 OWL Web Ontology Language Reference, <http://www.w3.org/TR/owl-ref/#OWLLite>
 OWL 웹 온톨로지 언어 개요 W3C 권고안 한글 번역문, W3C, <http://www.w3c.or.kr/Translation/REC-owl-features-20040210>
 protege, <http://protege.stanford.edu>
 RDF - RDF Schema - OWL references, W3C, <https://www.infowebml.ws/rdf-owl>
 1
 RDFS and OWL Classes, <https://www.infowebml.ws/rdf-owl/graphical-representations.htm#Statement>
 SILK, The Linked Data Integration Framework, <http://silkframework.org>

SKOS Simple Knowledge Organization System Namespace Document - HTML Variant, W3C, <http://www.w3.org/2004/02/skos/core#>

The Internet Engineering Task Force, <https://www.ietf.org>

Tim Berners-Lee, 「The next web」, TED 강연(온라인 참조, http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html)

Tim Berners-Lee, Linked Data, <http://www.w3.org/DesignIssues/LinkedData.html>

Time Ontology in OWL, W3C, <https://www.w3.org/TR/2017/WD-owl-time-20170202/#AF-97>

TopQuandant, <http://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition>

University of Sheffield, <https://gate.ac.uk>

UWordMap, 울산대학교 한국어처리연구실, <http://nlplab.ulsan.ac.kr/doku.php?id=uwordmap>

Validation Service, W3C, <https://www.w3.org/RDF/Validator>

VICODI project, <http://www.vicodi.org>

VOWL: Visual Notation for OWL Ontologies, <http://vowl.visualdataweb.org/protege/owl.html>

WebProtege 2.6.0, <https://github.com/protegeproject/webprotege/releases/tag/v2.6.0>

WordNet RDF, Princeton University, <http://wordnet-rdf.princeton.edu>

WordNet, Princeton University, <https://wordnet.princeton.edu>

World Map, Harvard University, <http://worldmap.harvard.edu>

YAGO, Max Planck Institute for Informatics, <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago>

【영문초록】

A Study on the Establishment of a Historical Text Knowledge Network through Digital Technology

- Focusing on the *Essentials of the Comprehensive Mirror (Tonggamjeoryo)* and its Annotated Translation

Chong Woong YOON
Major in Cultural Informatics
The Graduate School of Korean Studies
The Academy of Korean Studies

This dissertation aims to connect and integrate historical knowledge using linked open data (LOD). In Korea, a vast amount of historical material has been digitized through informatization projects and most of the resulting data can be utilized without any restrictions. However, despite this immense accumulation of data, those searching for relevant information end up wandering out into a sea of information since a large parts of the data are isolated from one another. To solve such problems, World Wide Web creator Tim Berners-Lee proposed the Semantic Web. The Semantic Web explicitly shows the significance of relationships between entities. On the Web, new facts are discovered and general knowledge can be acquired by deducing the dynamics between various entities by looking at their relationships to one another. LOD is one of the concrete ways to realize the Semantic Web. By publishing LOD, we can identify, connect, structure, and contextualize hidden knowledge to create meaning. However, LOD publication is currently concentrated in the areas of regular and standardized information. In other words, data transformed into LOD is mostly that which can be stored in standardized relational databases such as addresses or book metadata. Little effort has yet been made to publish LOD on the detailed content of historical texts.

This dissertation, which attempts to express the content of historical texts in LOD, consists of the following. First, the theoretical background and technical conditions for expressing historical knowledge is researched. Second, ways to express complex historical knowledge in an RDF triple is explored. Expressing historical knowledge with simple binary relations of RDF is no easy task. This challenge was handled by incorporating the concept of thematic relations used in syntax as a means to connect entities to one another. Third, LOD and an ontology model for the data was proposed as a method for connecting and integrating historical knowledge. To design an ontology model ad-

equate for historical data, the key entities of the entire data of the Essentials of the Comprehensive Mirror (通鑑節要) were identified and data of the Five Dynasties and Ten Kingdoms period was transformed into LOD to validate the adequacy of the ontology model. Fourth, a knowledge network was proposed to connect historical knowledge more organically and in depth. The knowledge network connects one knowledge node with other relevant nodes to connect otherwise segmented information. A large part of this knowledge can be connected to each other by publishing historical knowledge as LOD, however, it will be difficult for the connections to reach in-depth knowledge. Consequently, knowledge that has been refined by scholars, like that in dictionaries, also needs to be published in LOD. This kind of refined knowledge can be used to meaningfully connect entities to one another as synonyms, antonyms, or in hierarchical relationships, while events and entities must also be connected in terms of content.

In summary, the significance of this research is as follows. Until now, knowledge has largely been expressed in the form of standardized data. However, consumers now demand a more complex level of knowledge which cannot be found by merely connecting isolated pieces of knowledge. In this sense, the research presented in this dissertation is meaningful in its attempt to integrate and propose a model for expressing complex knowledge found in historical texts.

【부록】 historicevent 온톨로지

```
<?xml version="1.0"?>
```

```
<!DOCTYPE rdf:RDF [
```

```
<!ENTITY dcterms "http://purl.org/dc/terms/" >
```

```
<!ENTITY gml "http://www.opengis.net/gml#" >
```

```
<!ENTITY owl "http://www.w3.org/2002/07/owl#" >
```

```
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
```

```
<!ENTITY skos "http://www.w3.org/2004/02/skos/core#" >
```

```
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
```

```
<!ENTITY geo "http://www.w3.org/2003/01/geo/wgs84_pos#" >
```

```
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
```

```
<!ENTITY kno "http://dh.aks.ac.kr/ontologies/historicevent/kno#" >
```

```
<!ENTITY knod "http://dh.aks.ac.kr/ontologies/historicevent/knod#" >
```

```
<!ENTITY knoBook "http://dh.aks.ac.kr/ontologies/historicevent/Book/kno#" >
```

```
<!ENTITY knoPlace "http://dh.aks.ac.kr/ontologies/historicevent/Place/kno#" >
```

```
<!ENTITY knoPerson "http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#" >
```

```
<!ENTITY knoObject "http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#" >
```

```
<!ENTITY knoDynasty "http://dh.aks.ac.kr/ontologies/historicevent/Dynasty/kno#" >
```

```
<!ENTITY knoEraName "http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#" >
```

```
<!ENTITY knoPosition "http://dh.aks.ac.kr/ontologies/historicevent/Position/kno#" >
```

```
]>
```

```
<rdf:RDF xmlns="http://dh.aks.ac.kr/ontologies/historicevent#">
```

```
  xml:base="http://dh.aks.ac.kr/ontologies/historicevent"
```

```
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
```

```
  xmlns:knoDynasty="http://dh.aks.ac.kr/ontologies/historicevent/Dynasty/kno#">
```

```
  xmlns:knoPlace="http://dh.aks.ac.kr/ontologies/historicevent/Place/kno#">
```

```
  xmlns:knoObject="http://dh.aks.ac.kr/ontologies/historicevent/Object/kno#">
```

```
  xmlns:knoPosition="http://dh.aks.ac.kr/ontologies/historicevent/Position/kno#">
```

```
  xmlns:dcterms="http://purl.org/dc/terms/">
```

```
  xmlns:knoBook="http://dh.aks.ac.kr/ontologies/historicevent/Book/kno#">
```

```
  xmlns:knod="http://dh.aks.ac.kr/ontologies/historicevent/knod#">
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
  xmlns:owl="http://www.w3.org/2002/07/owl#">
```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:gml="http://www.opengis.net/gml#"
xmlns:kno="http://dh.aks.ac.kr/ontologies/historicevent/kno#"
xmlns:knoEraName="http://dh.aks.ac.kr/ontologies/historicevent/EraName/kno#"
xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:knoPerson="http://dh.aks.ac.kr/ontologies/historicevent/Person/kno#"
<owl:Ontology rdf:about="http://dh.aks.ac.kr/ontologies/historicevent">
  <rdfs:comment>통감절요를 대상으로 본문 중의 내용을 RDF로 기술하기 위한 온톨로지 설계이다. </rdfs:comment>
  <owl:versionIRI rdf:resource="http://dh.aks.ac.kr/ontologies/historicevent"/>
</owl:Ontology>

<!--
////////////////////////////////////
//
// Object Properties
//
////////////////////////////////////
-->

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#action -->
<owl:ObjectProperty rdf:about="&kno:action">
  <rdfs:label>행위</rdfs:label>
  <rdfs:comment>이벤트와 구체적인 행위 클래스와 연결한다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Action"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#after -->
<owl:ObjectProperty rdf:about="&kno:after">
  <rdfs:comment>A after B : A 뒤에 B가 발생하다(A를 기준으로)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Event"/>

```

```
<rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#agent -->
<owl:ObjectProperty rdf:about="&kno:agent">
  <rdfs:label>행동주</rdfs:label>
  <rdfs:comment>행동주</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Agent"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#beCausedBy -->
<owl:ObjectProperty rdf:about="&kno:beCausedBy">
  <rdfs:comment>~에서 유래하다
  원인이 되다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:causalRelationship"/>
  <owl:inverseOf rdf:resource="&kno:cause"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#before -->
<owl:ObjectProperty rdf:about="&kno:before">
  <rdfs:comment>A before B : B 이전에 A가 발생하다(B를 기준으로)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#birthPlace -->
<owl:ObjectProperty rdf:about="&kno:birthPlace">
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
```

```

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#bookInformation -->
<owl:ObjectProperty rdf:about="&kno:bookInformation">
  <rdfs:comment>서명과 관련된 정보를 표현한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Artifacts"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#capital -->
<owl:ObjectProperty rdf:about="&kno:capital">
  <rdfs:comment>왕조의 도읍을 표시함</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Dynasty"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:dynastyInfomation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#causalRelationship -->
<owl:ObjectProperty rdf:about="&kno:causalRelationship">
  <rdfs:comment>인과 관계를 표현</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#cause -->
<owl:ObjectProperty rdf:about="&kno:cause">
  <rdfs:label>원인</rdfs:label>
  <rdfs:comment>원인</rdfs:comment>
  <rdfs:comment>~ 결과를 가져오다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:causalRelationship"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#clan -->
<owl:ObjectProperty rdf:about="&kno:clan">
  <rdfs:comment>본관</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#companion -->
<owl:ObjectProperty rdf:about="&kno:companion">
  <rdfs:label>동반주</rdfs:label>
  <rdfs:comment>이벤트 행위를 할 때 참여한 행위자를 연결한다.
이벤트와 행위를 할 때 참여한 행위자를 가르킨다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#compose -->
<owl:ObjectProperty rdf:about="&kno:compose">
  <rdfs:comment>묘지명을撰한 관계</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#contains -->
<owl:ObjectProperty rdf:about="&kno:contains">
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#containsSubstance -->
<owl:ObjectProperty rdf:about="&kno:containsSubstance">

```

```

<rdfs:comment>자연물과 구성물질의 관계</rdfs:comment>
<rdfs:range rdf:resource="&kno:Object"/>
<rdfs:domain rdf:resource="&kno:Object"/>
<owl:inverseOf rdf:resource="&kno:isSubstanceOf"/>
<rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#content -->
<owl:ObjectProperty rdf:about="&kno:content">
  <rdfs:range rdf:resource="&kno:Content"/>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#continuousAction -->
<owl:ObjectProperty rdf:about="&kno:continuousAction">
  <rdfs:comment>연속행위의 표현</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno:eventType"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#deathPlace -->
<owl:ObjectProperty rdf:about="&kno:deathPlace">
  <rdfs:comment>사망지</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#direction -->
<owl:ObjectProperty rdf:about="&kno:direction">
  <rdfs:label>방향</rdfs:label>
  <rdfs:comment>이벤트와 이동의 방향을 나타내는 어휘와 연결한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>

```

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#during -->

<owl:ObjectProperty rdf:about="&kno:during">

<rdfs:range rdf:resource="&kno:Event"/>

<rdfs:domain rdf:resource="&kno:Event"/>

<rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#dynastyEnd -->

<owl:ObjectProperty rdf:about="&kno:dynastyEnd">

<rdfs:comment>왕조가 종료된 시간</rdfs:comment>

<rdfs:domain rdf:resource="&kno:Dynasty"/>

<rdfs:range rdf:resource="&kno:EraNameYear"/>

<rdfs:range rdf:resource="&kno:PosthumousNameYear"/>

<rdfs:range rdf:resource="&kno:SexagenaryCycle"/>

<rdfs:range rdf:resource="&kno:TempleNameYear"/>

<rdfs:range rdf:resource="&kno:Year"/>

<rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#dynastyInfomation -->

<owl:ObjectProperty rdf:about="&kno:dynastyInfomation">

<rdfs:comment>왕조와 관련된 정보</rdfs:comment>

<rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#dynastyStart -->

<owl:ObjectProperty rdf:about="&kno:dynastyStart">

<rdfs:comment>왕조가 시작된 시간</rdfs:comment>

<rdfs:domain rdf:resource="&kno:Dynasty"/>

<rdfs:range rdf:resource="&kno:EraNameYear"/>

<rdfs:range rdf:resource="&kno:PosthumousNameYear"/>

<rdfs:range rdf:resource="&kno:SexagenaryCycle"/>

<rdfs:range rdf:resource="&kno:TempleNameYear"/>

```

<rdfs:range rdf:resource="&kno;Year"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#effector -->
<owl:ObjectProperty rdf:about="&kno;effector">
<rdfs:label>영향주</rdfs:label>
<rdfs:comment>이벤트와 영향주와 연결한다.</rdfs:comment>
<rdfs:range rdf:resource="&kno;Effector"/>
<rdfs:domain rdf:resource="&kno;Event"/>
<rdfs:subPropertyOf rdf:resource="&kno;eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#emperor -->
<owl:ObjectProperty rdf:about="&kno;emperor">
<rdfs:comment>황제</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Dynasty"/>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;dynastyInfomation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#engrave -->
<owl:ObjectProperty rdf:about="&kno;engrave">
<rdfs:comment>묘지명을 篆한 관계</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#equals -->
<owl:ObjectProperty rdf:about="&kno;equals">
<rdfs:domain rdf:resource="&kno;Event"/>
<rdfs:range rdf:resource="&kno;Event"/>
<rdfs:subPropertyOf rdf:resource="&kno;eventTimeRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eraEnd -->
<owl:ObjectProperty rdf:about="&kno:eraEnd">
  <rdfs:comment>하나의 연호가 끝나는 해, 서기년도를 표시한다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:EraName"/>
  <rdfs:range rdf:resource="&kno:EraNameYear"/>
  <rdfs:range rdf:resource="&kno:PosthumousNameYear"/>
  <rdfs:range rdf:resource="&kno:TempleNameYear"/>
  <rdfs:range rdf:resource="&kno:Year"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eraStart -->
<owl:ObjectProperty rdf:about="&kno:eraStart">
  <rdfs:comment>하나의 연호가 시작되는 해, 서기년도로 표시한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:EraName"/>
  <rdfs:range rdf:resource="&kno:EraNameYear"/>
  <rdfs:range rdf:resource="&kno:PosthumousNameYear"/>
  <rdfs:range rdf:resource="&kno:TempleNameYear"/>
  <rdfs:range rdf:resource="&kno:Year"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eventRoleObject -->
<owl:ObjectProperty rdf:about="&kno:eventRoleObject">
  <rdfs:comment>이벤트는 목적어와 의미역으로 연결되어 있음을 표현한다.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eventTimeRelation -->
<owl:ObjectProperty rdf:about="&kno:eventTimeRelation">
  <rdfs:comment>추상적인 시간적 범위로 이루어져있으며, 시작과 끝 그리고 기간을 가지고 있음 </rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>

```

```
<rdf:s:subPropertyOf rdf:resource="&kno;topObjectProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eventType -->
<owl:ObjectProperty rdf:about="&kno:eventType">
  <rdf:comment>사건의 유형을 표시함</rdf:comment>
  <rdf:s:subPropertyOf rdf:resource="&kno;topObjectProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#finishes -->
<owl:ObjectProperty rdf:about="&kno:finishes">
  <rdf:domain rdf:resource="&kno:Event"/>
  <rdf:range rdf:resource="&kno:Event"/>
  <rdf:s:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#finishedBy -->
<owl:ObjectProperty rdf:about="&kno:finishedBy">
  <rdf:range rdf:resource="&kno:Event"/>
  <rdf:domain rdf:resource="&kno:Event"/>
  <rdf:s:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#goal -->
<owl:ObjectProperty rdf:about="&kno:goal">
  <rdf:label>도착점</rdf:label>
  <rdf:comment>이벤트와 도착점, 목표와 연결한다.</rdf:comment>
  <rdf:domain rdf:resource="&kno:Event"/>
  <rdf:range rdf:resource="&kno:Place"/>
  <rdf:range rdf:resource="&kno:Position"/>
  <rdf:s:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasAncestor -->
<owl:ObjectProperty rdf:about="&kno:hasAncestor">
```

```

<rdfs:comment>A 조상이다. B(A는 B를 조상으로 가지다. 혹은 A의 조상은 B이다.)</rdfs:comment>
<owl:inverseOf rdf:resource="&kno:hasDescendant"/>
<owl:propertyDisjointWith rdf:resource="&kno:hasDescendant"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasRelatedFamily"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasAntonym -->
<owl:ObjectProperty rdf:about="&kno:hasAntonym">
  <rdf:type rdf:resource="&owl:SymmetricProperty"/>
  <rdfs:comment>반의어 관계</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Term"/>
  <rdfs:range rdf:resource="&kno:Term"/>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBastardDaughter -->
<owl:ObjectProperty rdf:about="&kno:hasBastardDaughter">
  <rdf:type rdf:resource="&owl:AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:comment>서녀(서자의 반대)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasDescendant"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBastardSon -->
<owl:ObjectProperty rdf:about="&kno:hasBastardSon">
  <rdf:type rdf:resource="&owl:AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:comment>서자</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasDescendant"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBirthFather -->
<owl:ObjectProperty rdf:about="&kno:hasBirthFather">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>생부를 가지다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
  <owl:propertyDisjointWith rdf:resource="&kno:hasFosterFather"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBirthMother -->
<owl:ObjectProperty rdf:about="&kno:hasBirthMother">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>생모, 족보에서 두 번 결혼한 경우에 어머니가 누구인지 확인하기 위하여 생모를 구분해 준다.</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBranch -->
<owl:ObjectProperty rdf:about="&kno:hasBranch">
  <rdfs:comment>분과 계통의 관계</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Term"/>
  <rdfs:range rdf:resource="&kno:Term"/>
  <owl:inverseOf rdf:resource="&kno:isBranchOf"/>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasBrothers -->
<owl:ObjectProperty rdf:about="&kno:hasBrothers">
  <rdfs:comment>형제 관계</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSibling"/>

```

```
<owl:propertyDisjointWith rdf:resource="&kno;hasSisters"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasCareer -->
<owl:ObjectProperty rdf:about="&kno;hasCareer">
  <rdfs:comment>관직/ 직업을 갖다
  </rdfs:comment>
  <rdfs:comment>관력 정보, 이력 정보</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Position"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasComponent -->
<owl:ObjectProperty rdf:about="&kno;hasComponent">
  <rdfs:comment>전체와 구성요소와의 관계
  </rdfs:comment>
  <rdfs:comment>五禮 hasComponent 戒律宗(五禮의 구성 요소 중 하나는 戒律宗이다.</rdfs:comment>
  <rdfs:range rdf:resource="&kno;Term"/>
  <rdfs:domain rdf:resource="&kno;Term"/>
  <owl:inverseOf rdf:resource="&kno;isComponentOf"/>
  <rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasConcubine -->
<owl:ObjectProperty rdf:about="&kno;hasConcubine">
  <rdfs:comment>첩</rdfs:comment>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedAffinity"/>
  <owl:propertyDisjointWith rdf:resource="&kno;hasWife"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasDaughter -->
<owl:ObjectProperty rdf:about="&kno;hasDaughter">
  <rdfs:type rdf:resource="&owl;AsymmetricProperty"/>
  <rdfs:type rdf:resource="&owl;IrreflexiveProperty"/>
```

```

<rdfs:comment>친 딸</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasDescendant"/>
<owl:propertyDisjointWith rdf:resource="&kno;hasSon"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasDaughterInLaw -->
<owl:ObjectProperty rdf:about="&kno;hasDaughterInLaw">
<rdfs:comment>며느리</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedAffinity"/>
<owl:propertyDisjointWith rdf:resource="&kno;hasWife"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasDescendant -->
<owl:ObjectProperty rdf:about="&kno;hasDescendant">
<rdfs:comment>혼인을 통해서 파생되는 가족관계</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedFamily"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasDisciple -->
<owl:ObjectProperty rdf:about="&kno;hasDisciple">
<rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
<rdfs:comment>A는 B를 문인으로 두었다</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSocial"/>
<owl:inverseOf rdf:resource="&kno;isDiscipleOf"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasEraName -->

```

```

<owl:ObjectProperty rdf:about="&kno;hasEraName">
  <rdfs:comment>연호
당 태의 연호는 貞觀이다</rdfs:comment>
  <rdfs:comment>연호가 가지는 기간을 표현
武德(당 고조의 연호) 처럼 일정한 기간을 가지는 연호를 표현한다.</rdfs:comment>
  <rdfs:range rdf:resource="&kno;EraName"/>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasEraYear -->
<owl:ObjectProperty rdf:about="&kno;hasEraYear">
  <rdfs:comment>연호를 가지다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;EraName"/>
  <rdfs:range rdf:resource="&kno;EraNameYear"/>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasTime-Span"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasEvent -->
<owl:ObjectProperty rdf:about="&kno;hasEvent">
  <rdfs:comment>사건을 가지다</rdfs:comment>
  <rdfs:range rdf:resource="&kno;Event"/>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFather -->
<owl:ObjectProperty rdf:about="&kno;hasFather">
  <rdfs:comment>아버지를 가지다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasAncestor"/>
  <owl:propertyDisjointWith rdf:resource="&kno;hasMother"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFatherInLaw -->
<owl:ObjectProperty rdf:about="&kno:hasFatherInLaw">
  <rdfs:comment>장인</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFosterDaughter -->
<owl:ObjectProperty rdf:about="&kno:hasFosterDaughter">
  <rdf:type rdf:resource="&owl:AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:comment>양녀, 입양한 딸</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasDescendant"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFosterFather -->
<owl:ObjectProperty rdf:about="&kno:hasFosterFather">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>계부를 가지다.
양부, 입양이 되었을 때 아버지를 말한다. 계부는 의붓아버지이므로 이를 구분한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFosterSon -->
<owl:ObjectProperty rdf:about="&kno:hasFosterSon">
  <rdf:type rdf:resource="&owl:AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:comment>입양한 아들, 양자</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>

```

```

<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasDescendant"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasHusband -->
<owl:ObjectProperty rdf:about="&kno;hasHusband">
<rdf:type rdf:resource="&owl;AsymmetricProperty"/>
<rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
<rdfs:comment>남편</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedAffinity"/>
<owl:inverseOf rdf:resource="&kno;hasWife"/>
<owl:propertyDisjointWith rdf:resource="&kno;hasWife"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasIngredient -->
<owl:ObjectProperty rdf:about="&kno;hasIngredient">
<rdfs:comment>인공물과 재료의 관계</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Object"/>
<rdfs:range rdf:resource="&kno;Object"/>
<owl:inverseOf rdf:resource="&kno;isIngredientOf"/>
<rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasInstance -->
<owl:ObjectProperty rdf:about="&kno;hasInstance">
<rdfs:comment>개념과 사례 관계

```

■ 개념은 공간을 차지하고 있는 실체가 아닌 것을 말한다.

■ 관계 속성값으로 일반주제어가 상위개념어가 되고 사례는 하위개념어가 된다.</rdfs:comment>

```

<rdfs:range rdf:resource="&kno;Book"/>
<rdfs:domain rdf:resource="&kno;Book"/>
<rdfs:domain rdf:resource="&kno;ExaminationPasser"/>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Prize"/>

```

```

<owl:inverseOf rdf:resource="&kno;isInstanceOf"/>
<rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasKind -->
<owl:ObjectProperty rdf:about="&kno;hasKind">
<rdfs:comment>종류</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Term"/>
<rdfs:range rdf:resource="&kno;Term"/>
<owl:inverseOf rdf:resource="&kno;isKindOf"/>
<rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasLegitimateDaughter -->
<owl:ObjectProperty rdf:about="&kno;hasLegitimateDaughter">
<rdfs:comment>적녀</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasDescendant"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasLegitimateSon -->
<owl:ObjectProperty rdf:about="&kno;hasLegitimateSon">
<rdfs:comment>적자</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasDescendant"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasMember -->
<owl:ObjectProperty rdf:about="&kno;hasMember">
<rdfs:comment>집단과 구성원의 관계</rdfs:comment>
<owl:inverseOf rdf:resource="&kno;isMemberOf"/>
<rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
<rdfs:range rdf:resource="&owl;Thing"/>

```

```

<rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasMother -->
<owl:ObjectProperty rdf:about="&kno:hasMother">
  <rdf:type rdf:resource="&owl:AsymmetricProperty"/>
  <rdf:type rdf:resource="&owl:IrreflexiveProperty"/>
  <rdfs:comment>어머니, 일반적인 의미의 어머니를 의미한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasMotherInLaw -->
<owl:ObjectProperty rdf:about="&kno:hasMotherInLaw">
  <rdfs:comment>장모</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPart -->
<owl:ObjectProperty rdf:about="&kno:hasPart">
  <rdfs:comment>A는 B의 일부를 갖는다.</rdfs:comment>
  <owl:inverseOf rdf:resource="&kno:isPartOf"/>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPeriod -->
<owl:ObjectProperty rdf:about="&kno:hasPeriod">
  <rdfs:comment>시대를 가진다.</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>

```

```
<rdf:s:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPlace -->
<owl:ObjectProperty rdf:about="&kno:hasPlace">
  <rdf:comment>장소와 관련 있음을 표현</rdf:comment>
  <rdf:s:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPosition -->
<owl:ObjectProperty rdf:about="&kno:hasPosition">
  <rdf:comment>출사하다
  관직에 나가다, 역임하다</rdf:comment>
  <rdf:range rdf:resource="&kno:Office"/>
  <rdf:domain rdf:resource="&kno:Person"/>
  <rdf:range rdf:resource="&kno:Position"/>
  <rdf:s:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPostEvent -->
<owl:ObjectProperty rdf:about="&kno:hasPostEvent">
  <rdf:comment>후행 사건을 갖는다</rdf:comment>
  <rdf:domain rdf:resource="&kno:Event"/>
  <rdf:range rdf:resource="&kno:Event"/>
  <rdf:s:subPropertyOf rdf:resource="&kno:hasEvent"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPreEvent -->
<owl:ObjectProperty rdf:about="&kno:hasPreEvent">
  <rdf:comment>선행 사건을 갖는다</rdf:comment>
  <rdf:domain rdf:resource="&kno:Event"/>
  <rdf:range rdf:resource="&kno:Event"/>
  <rdf:s:subPropertyOf rdf:resource="&kno:hasEvent"/>
</owl:ObjectProperty>
```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelate -->
<owl:ObjectProperty rdf:about="&kno;hasRelate">
  <rdfs:range rdf:resource="&owl;Thing"/>
  <rdfs:domain rdf:resource="&owl;Thing"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedAffinity -->
<owl:ObjectProperty rdf:about="&kno;hasRelatedAffinity">
  <rdfs:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:type rdf:resource="&owl;TransitiveProperty"/>
  <rdfs:comment>인척관계, 즉 혼인에 의해서 맺어진 관계를 말한다.</rdfs:comment>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedFamily"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedDynasty -->
<owl:ObjectProperty rdf:about="&kno;hasRelatedDynasty">
  <rdfs:comment>왕국, 왕조와의 관계</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelate"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedFamily -->
<owl:ObjectProperty rdf:about="&kno;hasRelatedFamily">
  <rdfs:comment>관련된 가족을 가지고 있다.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedPerson"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedOrganization -->
<owl:ObjectProperty rdf:about="&kno;hasRelatedOrganization">
  <rdfs:comment>어떤 관서나 조직과 관련이 있을 경우</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Agent"/>
  <rdfs:range rdf:resource="&kno;Organization"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>

```

```

</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedPerson -->
<owl:ObjectProperty rdf:about="&kno:hasRelatedPerson">
  <rdf:type rdf:resource="&owl:SymmetricProperty"/>
  <rdf:type rdf:resource="&owl:TransitiveProperty"/>
  <rdfs:comment>한 인물이 다른 인물과 관계를 가지다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedSibling -->
<owl:ObjectProperty rdf:about="&kno:hasRelatedSibling">
  <rdfs:comment>형제 관계</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedFamily"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRelatedSocial -->
<owl:ObjectProperty rdf:about="&kno:hasRelatedSocial">
  <rdf:type rdf:resource="&owl:SymmetricProperty"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedPerson"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasReligion -->
<owl:ObjectProperty rdf:about="&kno:hasReligion">
  <rdfs:comment>종교를 가지다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Religion"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSisters -->

```

```

<owl:ObjectProperty rdf:about="&kno;hasSisters">
  <rdfs:comment>자매 관계</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSibling"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSon -->
<owl:ObjectProperty rdf:about="&kno;hasSon">
  <rdfs:comment>친자, 친아들</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasDescendant"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSonInLaw -->
<owl:ObjectProperty rdf:about="&kno;hasSonInLaw">
  <rdfs:comment>사위</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedAffinity"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasStatus -->
<owl:ObjectProperty rdf:about="&kno;hasStatus">
  <rdfs:comment>신분과 지위를 아우르는 포괄적 개념으로 사용한다.
馮廷諤의 신분은 僕夫이다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Status"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasStepMother -->
<owl:ObjectProperty rdf:about="&kno;hasStepMother">
  <rdfs:type rdf:resource="&owl;AsymmetricProperty"/>

```

```

<rdf:type rdf:resource="&kno:IrreflexiveProperty"/>
<rdfs:comment>A는 B를 양모로 가지고 있다.</rdfs:comment>
<rdfs:range rdf:resource="&kno:Person"/>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasAncestor"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSubEvent -->
<owl:ObjectProperty rdf:about="&kno:hasSubEvent">
<rdfs:comment>하위 사건</rdfs:comment>
<rdfs:range rdf:resource="&kno:Event"/>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasEvent"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSuperEvent -->
<owl:ObjectProperty rdf:about="&kno:hasSuperEvent">
<rdfs:comment>상위 사건을 가지다</rdfs:comment>
<rdfs:range rdf:resource="&kno:Event"/>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasEvent"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSynonym -->
<owl:ObjectProperty rdf:about="&kno:hasSynonym">
<rdfs:comment>동의를 표시함</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Term"/>
<rdfs:range rdf:resource="&kno:Term"/>
<rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasTime-Span -->
<owl:ObjectProperty rdf:about="&kno:hasTime-Span">
<rdfs:comment>기간을 가지다</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>

```

```

<rdfs:range rdf:resource="&kno;Time"/>
<rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasWife -->
<owl:ObjectProperty rdf:about="&kno;hasWife">
<rdfs:type rdf:resource="&owl;AsymmetricProperty"/>
<rdfs:type rdf:resource="&owl;IrreflexiveProperty"/>
<rdfs:comment>정처, 부인</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedAffinity"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasWorks -->
<owl:ObjectProperty rdf:about="&kno;hasWorks">
<rdfs:comment>저술, 저작, 작품을 쓰다</rdfs:comment>
<rdfs:range rdf:resource="&kno;Artifacts"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#instrument -->
<owl:ObjectProperty rdf:about="&kno;instrument">
<rdfs:label>이벤트와 도구를 나타나는 어휘와 연결한다.</rdfs:label>
<rdfs:comment>도구</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Event"/>
<rdfs:range rdf:resource="&kno;Object"/>
<rdfs:subPropertyOf rdf:resource="&kno;eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isBranchOf -->
<owl:ObjectProperty rdf:about="&kno;isBranchOf">
<rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
<rdfs:range rdf:resource="&owl;Thing"/>

```

```

<rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isComponentOf -->
<owl:ObjectProperty rdf:about="&kno:isComponentOf">
  <rdfs:comment>부분과 전체의 관계
  戒律宗 isComponentOf 五教兩宗
  戒律宗은 五教兩宗 중의 일부이다</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isDiscipleOf -->
<owl:ObjectProperty rdf:about="&kno:isDiscipleOf">
  <rdfs:comment>A는 B의 문인이다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isFriendOf -->
<owl:ObjectProperty rdf:about="&kno:isFriendOf">
  <rdfs:comment>A와 B는 친구이다</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isIngredientOf -->
<owl:ObjectProperty rdf:about="&kno:isIngredientOf">
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isInstanceOf -->
<owl:ObjectProperty rdf:about="&kno:isInstanceOf">
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isKindOf -->
<owl:ObjectProperty rdf:about="&kno:isKindOf">
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isMemberOf -->
<owl:ObjectProperty rdf:about="&kno:isMemberOf">
  <rdfs:comment>집단과 구성원의 관계
  </rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Organization"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

```

■ 일반 주체어와 人名, 團體名, 組織名 등의 특정 固有名稱이 구성원인 관계를 말한다.</rdfs:comment>

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isPartOf -->
<owl:ObjectProperty rdf:about="&kno:isPartOf">
  <rdfs:comment>A는 B의 일부이다.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
  <rdfs:range rdf:resource="&owl:Thing"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isRelatedTo -->
<owl:ObjectProperty rdf:about="&kno:isRelatedTo">
  <rdfs:type rdf:resource="&owl:SymmetricProperty"/>

```

```

<rdfs:comment>구체적으로 관계를 표현하기는 어려우나 관계가 있음을 표현한다.</rdfs:comment>
<rdfs:range rdf:resource="&kno;Person"/>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSocial"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#isSubstanceOf -->
<owl:ObjectProperty rdf:about="&kno;isSubstanceOf">
  <rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
  <rdfs:range rdf:resource="&owl;Thing"/>
  <rdfs:domain rdf:resource="&owl;Thing"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#knows -->
<owl:ObjectProperty rdf:about="&kno;knows">
  <rdfs:comment>서로 안면이 있는 관계</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasRelatedSocial"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#livingPlace -->
<owl:ObjectProperty rdf:about="&kno;livingPlace">
  <rdfs:comment>거주지</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:range rdf:resource="&kno;Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno;hasPlace"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#location -->
<owl:ObjectProperty rdf:about="&kno;location">
  <rdfs:label>처소</rdfs:label>
  <rdfs:comment>이벤트가 발생한 공간과 연결한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Event"/>
  <rdfs:range rdf:resource="&kno;Place"/>

```

```
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#master -->
<owl:ObjectProperty rdf:about="&kno:master">
  <rdfs:comment>주인</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
  <owl:inverseOf rdf:resource="&kno:servant"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#meets -->
<owl:ObjectProperty rdf:about="&kno:meets">
  <rdfs:comment>A는 B와 시간적으로 인접해 있다(A를 기준으로)</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#metBy -->
<owl:ObjectProperty rdf:about="&kno:metBy">
  <rdfs:comment>B는 A는 시간적으로 인접해 있다.B를 기준으로) </rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#nationality -->
<owl:ObjectProperty rdf:about="&kno:nationality">
  <rdfs:comment>국적</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Dynasty"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>
```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#nativePlace -->
<owl:ObjectProperty rdf:about="&kno:nativePlace">
  <rdfs:comment>출신지</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#occurPlace -->
<owl:ObjectProperty rdf:about="&kno:occurPlace">
  <rdfs:comment>활동지</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#overlappedBy -->
<owl:ObjectProperty rdf:about="&kno:overlappedBy">
  <rdfs:comment>A overlappedBy B : B가 A에 중첩된다.(B를 기준으로)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#overlaps -->
<owl:ObjectProperty rdf:about="&kno:overlaps">
  <rdfs:comment>A overlaps B : A가 B와 중첩된다(A를 기준으로 표현)</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#patient -->
<owl:ObjectProperty rdf:about="&kno:patient">

```

```

<rdfs:label>피동주</rdfs:label>
<rdfs:comment>이벤트와 피동주와 연결한다.
피동주는 사건에서 영향을 입는 것, 상태의 변화를 겪는 것</rdfs:comment>
<rdfs:range rdf:resource="&kno:Agent"/>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#penalColony -->
<owl:ObjectProperty rdf:about="&kno:penalColony">
<rdfs:comment>유배지</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#prisonPlace -->
<owl:ObjectProperty rdf:about="&kno:prisonPlace">
<rdfs:comment>수형지, 유배지</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#receiveAward -->
<owl:ObjectProperty rdf:about="&kno:receiveAward">
<rdfs:domain rdf:resource="&kno:Group"/>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:range rdf:resource="&kno:Prize"/>
<rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#reignEnd -->
<owl:ObjectProperty rdf:about="&kno:reignEnd">
<rdfs:comment>재위 끝</rdfs:comment>

```

```

<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#reignStart -->
<owl:ObjectProperty rdf:about="&kno:reignStart">
<rdfs:comment>재위년 시작</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#route -->
<owl:ObjectProperty rdf:about="&kno:route">
<rdfs:label>경로</rdfs:label>
<rdfs:comment>이벤트와 경로를 연결한다.</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#sameYear -->
<owl:ObjectProperty rdf:about="&kno:sameYear">
<rdfs:comment>333</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Dynasty"/>
<rdfs:domain rdf:resource="&kno:EraName"/>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:domain rdf:resource="&kno:PosthumousName"/>
<rdfs:domain rdf:resource="&kno:TempleName"/>
<rdfs:range rdf:resource="&kno:Year"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#servant -->
<owl:ObjectProperty rdf:about="&kno:servant">
<rdfs:comment>종</rdfs:comment>

```

```

<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:range rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#sexagenarycycle -->
<owl:ObjectProperty rdf:about="&kno:sexagenarycycle">
<rdfs:domain rdf:resource="&kno:Day"/>
<rdfs:domain rdf:resource="&kno:Month"/>
<rdfs:range rdf:resource="&kno:SexagenaryCycle"/>
<rdfs:domain rdf:resource="&kno:Year"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasTime-Span"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#source -->
<owl:ObjectProperty rdf:about="&kno:source">
<rdfs:label>출발점</rdfs:label>
<rdfs:comment>이벤트와 출발점을 연결한다.</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:range rdf:resource="&kno:Position"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#spatiallyInclude -->
<owl:ObjectProperty rdf:about="&kno:spatiallyInclude">
<rdfs:comment>공간적 부분 관계
唐 spatiallyInclude 京畿道(당나라에는 경기도를 포함하고 있다.)</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Dynasty"/>
<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:domain rdf:resource="&kno:Place"/>
<owl:inverseOf rdf:resource="&kno:spatiallyInside"/>
<rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#spatiallyInside -->
<owl:ObjectProperty rdf:about="&kno:spatiallyInside">
  <rdfs:comment>京畿道 spatiallyInside 唐(京畿道는 당나라에 속해있다.)</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Dynasty"/>
  <rdfs:range rdf:resource="&kno:Place"/>
  <rdfs:domain rdf:resource="&kno:Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#startedBy -->
<owl:ObjectProperty rdf:about="&kno:startedBy">
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#starts -->
<owl:ObjectProperty rdf:about="&kno:starts">
  <rdfs:range rdf:resource="&kno:Event"/>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventTimeRelation"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#subordinate -->
<owl:ObjectProperty rdf:about="&kno:subordinate">
  <rdfs:comment>부하</rdfs:comment>
  <rdfs:range rdf:resource="&kno:Person"/>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
  <owl:inverseOf rdf:resource="&kno:superior"/>
</owl:ObjectProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#superior -->
<owl:ObjectProperty rdf:about="&kno:superior">
  <rdfs:comment>상사, 주인을 표현한다.</rdfs:comment>

```

```

<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:range rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#termRelation -->
<owl:ObjectProperty rdf:about="&kno:termRelation">
<rdfs:comment>어휘 간 관계를 표현함</rdfs:comment>
<rdfs:range rdf:resource="&owl:Thing"/>
<rdfs:domain rdf:resource="&owl:Thing"/>
<rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#theme -->
<owl:ObjectProperty rdf:about="&kno:theme">
<rdfs:label>대상</rdfs:label>
<rdfs:comment>이벤트와 대상을 연결한다.</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:range rdf:resource="&kno:Event"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#time -->
<owl:ObjectProperty rdf:about="&kno:time">
<rdfs:label>시간</rdfs:label>
<rdfs:comment>이벤트와 시간을 연결한다.</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:range rdf:resource="&kno:Time"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleObject"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#tombPlace -->
<owl:ObjectProperty rdf:about="&kno:tombPlace">
<rdfs:comment>묘, 장지위치</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>

```

```

<rdfs:range rdf:resource="&kno:Place"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasPlace"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#translationWord -->
<owl:ObjectProperty rdf:about="&kno:translationWord">
<rdf:type rdf:resource="&owl:SymmetricProperty"/>
<rdfs:subPropertyOf rdf:resource="&kno:termRelation"/>
<rdfs:domain rdf:resource="&owl:Thing"/>
<rdfs:range rdf:resource="&owl:Thing"/>
</owl:ObjectProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#writes -->
<owl:ObjectProperty rdf:about="&kno:writes">
<rdfs:comment>A는 B를 쓴다
묘지명을 사용한 관계</rdfs:comment>
<rdfs:range rdf:resource="&kno:Person"/>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:hasRelatedSocial"/>
</owl:ObjectProperty>

<!-- http://purl.org/dc/terms/creator -->
<owl:ObjectProperty rdf:about="&dcterms:creator">
<rdfs:comment>A의 저자는 B이다. </rdfs:comment>
<rdfs:range rdf:resource="&kno:Agent"/>
<rdfs:domain rdf:resource="&kno:Artifacts"/>
<rdfs:subPropertyOf rdf:resource="&kno:bookInformation"/>
</owl:ObjectProperty>

<!-- http://purl.org/dc/terms/title -->
<owl:ObjectProperty rdf:about="&dcterms:title">
<rdfs:comment>A책의 제목은 B이다
일반적으로 그 자원에 대해서 공식적으로 알려진 이름이다.</rdfs:comment>
<rdfs:range rdf:resource="&kno:Agent"/>
<rdfs:domain rdf:resource="&kno:Artifacts"/>

```

```
<rdfs:subPropertyOf rdf:resource="&kno;bookInformation"/>
</owl:ObjectProperty>
```

```
<!-- http://www.w3.org/2002/07/owl#topObjectProperty -->
<owl:ObjectProperty rdf:about="&owl;topObjectProperty">
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
</owl:ObjectProperty>
```

```
<!-- http://www.w3.org/2004/02/skos/core#broader -->
<owl:ObjectProperty rdf:about="&skos;broader">
  <rdfs:comment>상위어</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
  <rdfs:domain rdf:resource="&owl;Thing"/>
  <rdfs:range rdf:resource="&owl;Thing"/>
  <owl:inverseOf rdf:resource="&skos;narrower"/>
</owl:ObjectProperty>
```

```
<!-- http://www.w3.org/2004/02/skos/core#narrower -->
<owl:ObjectProperty rdf:about="&skos;narrower">
  <rdfs:comment>상위개념어</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&kno;termRelation"/>
  <rdfs:range rdf:resource="&owl;Thing"/>
  <rdfs:domain rdf:resource="&owl;Thing"/>
</owl:ObjectProperty>
```

```
<!--
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
-->
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#UCI -->
<owl:DatatypeProperty rdf:about="&kno;UCI">
```

```

<rdfs:comment>고유값, 한중연 인물 정보와 연계 시키기 위함</rdfs:comment>
<rdfs:subPropertyOf rdf:resource="&kno;personId"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#WebResource -->
<owl:DatatypeProperty rdf:about="&kno;WebResource">
<rdfs:label xml:lang="ko">웹 정보자원</rdfs:label>
<rdfs:comment xml:lang="ko">웹 서비스 페이지 등 웹 상에서 표현되고, URL을 가지고 있는 정보 자원의
개념이다.</rdfs:comment>
<rdfs:range rdf:resource="&rdf:XMLLiteral"/>
<rdfs:domain rdf:resource="&owl:Thing"/>
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#address -->
<owl:DatatypeProperty rdf:about="&kno;address">
<rdfs:comment>행정구역 지명을 지칭한다.
河南道宋州陽山縣午溝里</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Place"/>
<rdfs:range rdf:resource="&xsd:string"/>
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#anotherName -->
<owl:DatatypeProperty rdf:about="&kno;anotherName">
<rdfs:comment>이칭, 별칭, 사람인명을 제외한 모든 명칭에 사용한다.</rdfs:comment>
<rdfs:range rdf:resource="&rdfs:Literal"/>
<rdfs:domain rdf:resource="&owl:Thing"/>
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#birthDate -->
<owl:DatatypeProperty rdf:about="&kno;birthDate">
<rdfs:comment>사람이 태어난 날짜</rdfs:comment>

```

```

<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:dateInformation"/>
<rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#complement -->
<owl:DatatypeProperty rdf:about="&kno:complement">
<rdfs:comment>기타 보어</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
<rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#dateInformation -->
<owl:DatatypeProperty rdf:about="&kno:dateInformation">
<rdfs:comment>날짜</rdfs:comment>
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#deathDate -->
<owl:DatatypeProperty rdf:about="&kno:deathDate">
<rdfs:comment>사람이 사망한 날짜</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:dateInformation"/>
<rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#degree -->
<owl:DatatypeProperty rdf:about="&kno:degree">
<rdfs:label>정도</rdfs:label>
<rdfs:comment>이벤트와 행위나 상태의 정도 나타내는 어휘와 연결한다.
</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Event"/>
<rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
<rdfs:range rdf:resource="&rdfs:Literal"/>

```

```

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#description -->
<owl:DatatypeProperty rdf:about="&kno:description">
  <rdfs:comment>어휘에 대한 간단한 설명, definition이 정제화된 정의를 의미한다면 어휘에 대한 설명은 description을 사용한다.</rdfs:comment>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#distance -->
<owl:DatatypeProperty rdf:about="&kno:distance">
  <rdfs:comment>어떤 지점과 어느 정도의 거리가 떨어져 있다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#endDate -->
<owl:DatatypeProperty rdf:about="&kno:endDate">
  <rdfs:comment>사건, 수행, 종교, 관직 등이 완료된 시간</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:dateInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#etcsubject -->
<owl:DatatypeProperty rdf:about="&kno:etcsubject">
  <rdfs:domain rdf:resource="&kno:Agent"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#eventRoleData -->

```

```
<owl:DatatypeProperty rdf:about="&kno:eventRoleData">
  <rdfs:comment>Datatype 의미역을 표현함</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#familyClan -->
<owl:DatatypeProperty rdf:about="&kno:familyClan">
  <rdfs:comment>본관을</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#familyName -->
<owl:DatatypeProperty rdf:about="&kno:familyName">
  <rdfs:comment>성씨를 표현한다.</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personInformation"/>
</owl:DatatypeProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#frequency -->
<owl:DatatypeProperty rdf:about="&kno:frequency">
  <rdfs:comment>빈도</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#gender -->
<owl:DatatypeProperty rdf:about="&kno:gender">
  <rdfs:comment>성별, 男, 女로 구분한다</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>
```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#geoInformation -->
<owl:DatatypeProperty rdf:about="&kno;geoInformation">
  <rdfs:comment>공간 위치 정보</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Place"/>
  <rdfs:range rdf:resource="&xsd:string"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#gisURL -->
<owl:DatatypeProperty rdf:about="&kno;gisURL">
  <rdfs:domain rdf:resource="&kno;Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno;geoInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#governmentComingDate -->
<owl:DatatypeProperty rdf:about="&kno;governmentComingDate">
  <rdfs:comment>출사일</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;dateInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasAliasName -->
<owl:DatatypeProperty rdf:about="&kno;hasAliasName">
  <rdfs:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>이칭</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasChangedName -->
<owl:DatatypeProperty rdf:about="&kno;hasChangedName">

```

```

<rdf:type rdf:resource="&owl:FunctionalProperty"/>
<rdfs:comment>가명</rdfs:comment>
<rdfs:domain rdf:resource="&kno:Person"/>
<rdfs:subPropertyOf rdf:resource="&kno:personName"/>
<rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasChildhoodName -->
<owl:DatatypeProperty rdf:about="&kno:hasChildhoodName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>아명</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasCourtesyName -->
<owl:DatatypeProperty rdf:about="&kno:hasCourtesyName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>자</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasDharmaName -->
<owl:DatatypeProperty rdf:about="&kno:hasDharmaName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>법호, 법명</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasEnfoeffedName -->

```

```

<owl:DatatypeProperty rdf:about="&kno;hasEnfoeffedName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>봉호, 봉작호, 왕이 봉하여 내려 준 호(號)를 말한다. </rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasFormerName -->
<owl:DatatypeProperty rdf:about="&kno;hasFormerName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>초명</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasMeritoriousOfficials -->
<owl:DatatypeProperty rdf:about="&kno;hasMeritoriousOfficials">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>공신호</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasName -->
<owl:DatatypeProperty rdf:about="&kno;hasName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>이름이면서 특정하게 분류하기 어려운 경우</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasNickName -->
<owl:DatatypeProperty rdf:about="&kno:hasNickName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>호, 아호</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasOfficialName -->
<owl:DatatypeProperty rdf:about="&kno:hasOfficialName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>공식명칭, 대표명칭</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPenName -->
<owl:DatatypeProperty rdf:about="&kno:hasPenName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>필명</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasPosthumousName -->
<owl:DatatypeProperty rdf:about="&kno:hasPosthumousName">
  <rdf:type rdf:resource="&owl:FunctionalProperty"/>
  <rdfs:comment>시호</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:personName"/>
  <rdfs:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasRealName -->
<owl:DatatypeProperty rdf:about="&kno;hasRealName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>본명, 가명이나 별명 등에 대해서 본래 이름을 말함 </rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasReligionName -->
<owl:DatatypeProperty rdf:about="&kno;hasReligionName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>종교명(이름연관된)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasSpellingVariations -->
<owl:DatatypeProperty rdf:about="&kno;hasSpellingVariations">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>철자 이행 이름</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#hasTempleName -->
<owl:DatatypeProperty rdf:about="&kno;hasTempleName">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:comment>묘호</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personName"/>
  <rdfs:range rdf:resource="&xsd;language"/>

```

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#honorDate -->

<owl:DatatypeProperty rdf:about="&kno:honorDate">

<rdfs:comment>추상일</rdfs:comment>

<rdfs:domain rdf:resource="&kno:Person"/>

<rdfs:subPropertyOf rdf:resource="&kno:dateInformation"/>

<rdfs:range rdf:resource="&rdfs:Literal"/>

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#identifier -->

<owl:DatatypeProperty rdf:about="&kno:identifier">

<rdfs:comment>외부 자원의 ID</rdfs:comment>

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#jtknowIdentifier -->

<owl:DatatypeProperty rdf:about="&kno;jtknowIdentifier">

<rdfs:comment>전통문화연구회의 어휘DB의 ID</rdfs:comment>

<rdfs:subPropertyOf rdf:resource="&kno:identifier"/>

<rdfs:range rdf:resource="&rdfs:Literal"/>

<rdfs:domain rdf:resource="&owl:Thing"/>

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#legitimacy -->

<owl:DatatypeProperty rdf:about="&kno:legitimacy">

<rdfs:comment>적서구분, 처첩구분,嫡,庶</rdfs:comment>

<rdfs:domain rdf:resource="&kno:Person"/>

<rdfs:subPropertyOf rdf:resource="&kno:personInformation"/>

<rdfs:range rdf:resource="&rdfs:Literal"/>

</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#locationType -->

<owl:DatatypeProperty rdf:about="&kno:locationType">

<rdfs:comment>Location의 유형, 전, 후, 좌, 우, 근처 등이 올 수 있다.</rdfs:comment>

<rdfs:domain rdf:resource="&kno:Event"/>

```

<rdf:s:subPropertyOf rdf:resource="&kno:eventRoleData"/>
<rdf:s:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#mausoleum -->
<owl:DatatypeProperty rdf:about="&kno:mausoleum">
<rdf:s:comment>능묘</rdf:s:comment>
<rdf:s:domain rdf:resource="&kno:Person"/>
<rdf:s:subPropertyOf rdf:resource="&kno:personInformation"/>
<rdf:s:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#order -->
<owl:DatatypeProperty rdf:about="&kno:order">
<rdf:s:domain rdf:resource="&kno:Agent"/>
<rdf:s:subPropertyOf rdf:resource="&kno:relationshipOrder"/>
<rdf:s:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#personInformation -->
<owl:DatatypeProperty rdf:about="&kno:personInformation">
<rdf:s:comment rdf:datatype="&rdfs:Literal">인명과 관련된 정보</rdf:s:comment>
<rdf:s:domain rdf:resource="&kno:Person"/>
<rdf:s:range rdf:resource="&xsd:language"/>
<rdf:s:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#personName -->
<owl:DatatypeProperty rdf:about="&kno:personName">
<rdf:s:comment>인물 이름</rdf:s:comment>
<rdf:s:domain rdf:resource="&kno:Person"/>
<rdf:s:subPropertyOf rdf:resource="&kno:personInformation"/>
<rdf:s:range rdf:resource="&xsd:language"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#pesonId -->
<owl:DatatypeProperty rdf:about="&kno;pesonId">
  <rdfs:comment>인물 ID</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno;personInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#polygon -->
<owl:DatatypeProperty rdf:about="&kno;polygon">
  <rdfs:comment>공간 정보에서 면을 의미</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno;geoInformation"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#polyline -->
<owl:DatatypeProperty rdf:about="&kno;polyline">
  <rdfs:comment>공간 정보 중 선으로 되어 있는 것으로 복수의 경위도 정보가 올 수 있음</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Place"/>
  <rdfs:subPropertyOf rdf:resource="&kno;geoInformation"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#positionInformation -->
<owl:DatatypeProperty rdf:about="&kno;positionInformation">
  <rdfs:comment>관직</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Position"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#positionType -->
<owl:DatatypeProperty rdf:about="&kno;positionType">
  <rdfs:comment>관직유형(문반/무반)</rdfs:comment>
  <rdfs:domain rdf:resource="&kno;Position"/>

```

```

<rdfs:subPropertyOf rdf:resource="&kno;positionInformation"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#quantity -->
<owl:DatatypeProperty rdf:about="&kno;quantity">
<rdfs:comment>양</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Event"/>
<rdfs:subPropertyOf rdf:resource="&kno;eventRoleData"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#rank -->
<owl:DatatypeProperty rdf:about="&kno;rank">
<rdfs:comment>품계</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Position"/>
<rdfs:subPropertyOf rdf:resource="&kno;positionInformation"/>
<rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#recordOrder -->
<owl:DatatypeProperty rdf:about="&kno;recordOrder">
<rdfs:comment>등계순서</rdfs:comment>
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:subPropertyOf rdf:resource="&kno;relationshipOrder"/>
<rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#relationshipOrder -->
<owl:DatatypeProperty rdf:about="&kno;relationshipOrder">
<rdfs:domain rdf:resource="&kno;Person"/>
<rdfs:range rdf:resource="&xsd:int"/>
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#role -->
<owl:DatatypeProperty rdf:about="&kno;role">
  <rdfs:domain rdf:resource="&kno:Event"/>
  <rdfs:subPropertyOf rdf:resource="&kno:eventRoleData"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#siblingOrder -->
<owl:DatatypeProperty rdf:about="&kno;siblingOrder">
  <rdfs:comment>형제순서</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:relationshipOrder"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#spouseOrder -->
<owl:DatatypeProperty rdf:about="&kno;spouseOrder">
  <rdfs:comment>배우자순서</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:relationshipOrder"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#startDate -->
<owl:DatatypeProperty rdf:about="&kno;startDate">
  <rdfs:comment>사건, 수행, 종교, 관직 등이 발생 혹은 시작된 시간, 연호의 시작된 때</rdfs:comment>
  <rdfs:domain rdf:resource="&kno:Person"/>
  <rdfs:subPropertyOf rdf:resource="&kno:dateInformation"/>
  <rdfs:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#stringId -->
<owl:DatatypeProperty rdf:about="&kno:stringId">
  <rdfs:comment>문자열로 구성된 식별자로 id가 가독성이 떨어지기 때문에 이를 보완할 목적으로 생성함</r
dfs:comment>

```

```

<rdf:s:subPropertyOf rdf:resource="&kno;pesonId"/>
</owl:DatatypeProperty>

<!-- http://purl.org/dc/terms/created -->
<owl:DatatypeProperty rdf:about="&dcterms:created">
  <rdf:comment>작성일, 생산일, 문서나 책을 작성한 날짜</rdf:comment>
  <rdf:domain rdf:resource="&kno;Artifacts"/>
  <rdf:s:subPropertyOf rdf:resource="&kno;dateInformation"/>
  <rdf:s:range rdf:resource="&rdfs:Literal"/>
</owl:DatatypeProperty>

<!-- http://www.w3.org/2002/07/owl#topDataProperty -->
<owl:DatatypeProperty rdf:about="&owl;topDataProperty"/>

<!-- http://www.w3.org/2003/01/geo/wgs84_pos#lat -->
<owl:DatatypeProperty rdf:about="&geo:lat">
  <rdf:domain rdf:resource="&kno;Place"/>
  <rdf:s:subPropertyOf rdf:resource="&kno;geoInformation"/>
  <rdf:s:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>

<!-- http://www.w3.org/2003/01/geo/wgs84_pos#long -->
<owl:DatatypeProperty rdf:about="&geo:long">
  <rdf:domain rdf:resource="&kno;Place"/>
  <rdf:s:subPropertyOf rdf:resource="&kno;geoInformation"/>
  <rdf:s:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>

<!-- http://www.w3.org/2004/02/skos/core#definition -->
<owl:DatatypeProperty rdf:about="&skos:definition">
  <rdf:label>definition</rdf:label>
  <rdf:comment rdf:datatype="&xsd:string">어휘의 定義, 사전에서 어휘를 정의하는 키워드에 적용.</rdf:c
omment>
  <rdf:s:range rdf:resource="&rdfs:Literal"/>
  <rdf:s:domain rdf:resource="&owl:Thing"/>

```

```
<rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
```

```
<!-- http://www.w3.org/2004/02/skos/core#note -->
<owl:DatatypeProperty rdf:about="&skos:note">
  <rdfs:label>note</rdfs:label>
  <rdfs:comment>비고, 기타 추가할 내용을 기록한다.</rdfs:comment>
  <rdfs:range rdf:resource="&xsd:string"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
```

```
<!-- http://www.w3.org/2004/02/skos/core#prefLabel -->
<owl:DatatypeProperty rdf:about="&skos:prefLabel">
  <rdfs:comment>대표명칭</rdfs:comment>
  <rdfs:range rdf:resource="&xsd:language"/>
  <rdfs:domain rdf:resource="&owl:Thing"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topDataProperty"/>
</owl:DatatypeProperty>
```

```
<!--
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
-->
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Action -->
<owl:Class rdf:about="&kno:Action">
  <rdfs:comment>행위</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Agent -->
<owl:Class rdf:about="&kno:Agent">
```

```

<rdfs:comment>행위의 주체, 인물, 기관</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Animal -->
<owl:Class rdf:about="&kno:Animal">
  <rdfs:comment>동물</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Artifacts -->
<owl:Class rdf:about="&kno:Artifacts">
  <rdfs:comment>인공 구조물</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#AstronomicalPhenomenon -->
<owl:Class rdf:about="&kno:AstronomicalPhenomenon">
  <rdfs:comment>천문 현상</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Book -->
<owl:Class rdf:about="&kno:Book">
  <rdfs:subClassOf rdf:resource="&kno:Artifacts"/>
  <rdfs:comment></rdfs:comment>
  <rdfs:comment>도서, 서적 작품명 등</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Chinese -->
<owl:Class rdf:about="&kno:Chinese">
  <rdfs:subClassOf rdf:resource="&kno:Action"/>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#ClassicsLicentiateExamination -->
<owl:Class rdf:about="&kno:ClassicsLicentiateExamination">
  <rdfs:subClassOf rdf:resource="&kno:ExaminationPasser"/>
  <rdfs:comment>생원진사시 합격자 정보
홍길동의 관거합격자 정보</rdfs:comment>

```

</owl:Class>

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Concept -->
<owl:Class rdf:about="&kno;Concept">
  <rdfs:comment>컨셉, 개념</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Content -->
<owl:Class rdf:about="&kno;Content">
  <rdfs:comment>문장 형태의 역사적 지식</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Day -->
<owl:Class rdf:about="&kno;Day">
  <rdfs:subClassOf rdf:resource="&kno;Time"/>
  <rdfs:comment>일, </rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Description -->
<owl:Class rdf:about="&kno;Description">
  <rdfs:comment>정의</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Document -->
<owl:Class rdf:about="&kno;Document">
  <rdfs:subClassOf rdf:resource="&kno;Artifacts"/>
  <rdfs:comment>문서</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Dynasty -->
<owl:Class rdf:about="&kno;Dynasty">
  <rdfs:comment>왕조 명칭</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Effector -->
```

```

<owl:Class rdf:about="&kno:Effector">
  <rdfs:subClassOf rdf:resource="&kno:Agent"/>
  <rdfs:comment>영향주, 동사가 나타내는 사건이나 형용사가 나타내는 현상을 비의도적으로 유발하는 주어
  를 말한다. 보통 無情物 주어인 경우가 많다.</rdfs:comment>
</owl:Class>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#EraName -->
<owl:Class rdf:about="&kno:EraName">
  <rdfs:subClassOf rdf:resource="&kno:Time-Span"/>
  <rdfs:comment>연호
  당 태의 연호는 貞觀이다</rdfs:comment>
  <rdfs:comment>연호가 가지는 기간을 표현
  武德(당 고조의 연호) 처럼 일정한 기간을 가지는 연호를 표현한다.</rdfs:comment>
</owl:Class>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#EraNameYear -->
<owl:Class rdf:about="&kno:EraNameYear">
  <rdfs:subClassOf rdf:resource="&kno:Time-Span"/>
  <rdfs:comment>연호와 숫자를 결합하여 연도를 나타내는 경우 貞觀1</rdfs:comment>
</owl:Class>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#EruditeExamination -->
<owl:Class rdf:about="&kno:EruditeExamination">
  <rdfs:subClassOf rdf:resource="&kno:ExaminationPasser"/>
  <rdfs:comment>문과합격자 정보</rdfs:comment>
</owl:Class>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Event -->
<owl:Class rdf:about="&kno:Event">
  <rdfs:comment>사건 정보 </rdfs:comment>
</owl:Class>

```

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Examination -->
<owl:Class rdf:about="&kno:Examination">
  <rdfs:subClassOf rdf:resource="&kno:System"/>

```

```

<rdfs:comment>과거 시험명</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#ExaminationPasser -->
<owl:Class rdf:about="&kno:ExaminationPasser">
  <rdfs:subClassOf rdf:resource="&kno:System"/>
  <rdfs:comment>과거합격자 명단</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Faction -->
<owl:Class rdf:about="&kno:Faction">
  <rdfs:subClassOf rdf:resource="&kno:Agent"/>
  <rdfs:comment>과벌, 당파, 분파, 도당</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Family -->
<owl:Class rdf:about="&kno:Family">
  <rdfs:subClassOf rdf:resource="&kno:Agent"/>
  <rdfs:comment>가문</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Group -->
<owl:Class rdf:about="&kno:Group">
  <rdfs:subClassOf rdf:resource="&kno:Agent"/>
  <rdfs:comment>집단, 조직, 당파, 학파 등의 관계를 의미한다.</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Idiom -->
<owl:Class rdf:about="&kno:Idiom">
  <rdfs:comment>고사성어</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Korean -->
<owl:Class rdf:about="&kno:Korean">
  <rdfs:subClassOf rdf:resource="&kno:Action"/>

```

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#MilitaryExamination -->

<owl:Class rdf:about="&kno;MilitaryExamination">

<rdfs:subClassOf rdf:resource="&kno;ExaminationPasser"/>

<rdfs:comment>무과합격자 정보</rdfs:comment>

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#MiscellaneousExamination -->

<owl:Class rdf:about="&kno;MiscellaneousExamination">

<rdfs:subClassOf rdf:resource="&kno;ExaminationPasser"/>

<rdfs:comment>잡과 합격자 정보</rdfs:comment>

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Month -->

<owl:Class rdf:about="&kno;Month">

<rdfs:subClassOf rdf:resource="&kno;Time"/>

<rdfs:comment>월</rdfs:comment>

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Object -->

<owl:Class rdf:about="&kno;Object">

<rdfs:comment>일반적인 사물을 의미한다</rdfs:comment>

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Office -->

<owl:Class rdf:about="&kno;Office">

<rdfs:comment>관서</rdfs:comment>

</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Organization -->

<owl:Class rdf:about="&kno;Organization">

<rdfs:subClassOf rdf:resource="&kno;Agent"/>

<rdfs:comment>기관 단체</rdfs:comment>

</owl:Class>

```

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Period -->
<owl:Class rdf:about="&kno:Period">
  <rdfs:subClassOf rdf:resource="&kno:Time"/>
  <rdfs:comment>시간과 공간에 제약된 일관성 있는 현상 또는 문화적 징후들의 합으로, 특히 선사시대 또는
역사시대를 기술하는데 사용됨
청동기 시대, 철기 시대 등</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Person -->
<owl:Class rdf:about="&kno:Person">
  <rdfs:subClassOf rdf:resource="&kno:Agent"/>
  <rdfs:comment>인물, 대표명칭(representativename)으로 표현한다.</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Place -->
<owl:Class rdf:about="&kno:Place">
  <rdfs:comment>지명 정보.</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Plant -->
<owl:Class rdf:about="&kno:Plant">
  <rdfs:comment>식물</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Position -->
<owl:Class rdf:about="&kno:Position">
  <rdfs:comment>관직</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#PosthumousName -->
<owl:Class rdf:about="&kno:PosthumousName">
  <rdfs:subClassOf rdf:resource="&kno:Time-Span"/>
  <rdfs:comment>襄公 처럼 일정한 기간을 가지고 있음
채위 기간을 가지고 있는 시간을 의미함

```

```

시호</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#PosthumousNameYear -->
<owl:Class rdf:about="&kno;PosthumousNameYear">
  <rdfs:subClassOf rdf:resource="&kno;Time-Span"/>
  <rdfs:comment>당대 이전 시호와 연도를 결합하여 연도를 나타내는 경우
  隱公 3年, 襄公 26年 등</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Prize -->
<owl:Class rdf:about="&kno;Prize">
  <rdfs:comment>상</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Religion -->
<owl:Class rdf:about="&kno;Religion">
  <rdfs:comment>종교</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#SexagenaryCycle -->
<owl:Class rdf:about="&kno;SexagenaryCycle">
  <rdfs:subClassOf rdf:resource="&kno;Time"/>
  <rdfs:comment>간지, 60갑자</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Source -->
<owl:Class rdf:about="&kno;Source">
  <rdfs:comment>출처 정보</rdfs:comment>
</owl:Class>

<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Status -->
<owl:Class rdf:about="&kno;Status">
  <rdfs:comment>법적, 사회적 신분</rdfs:comment>
</owl:Class>

```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#System -->
<owl:Class rdf:about="&kno;System">
  <rdfs:comment>제도/체제</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#TempleName -->
<owl:Class rdf:about="&kno;TempleName">
  <rdfs:subClassOf rdf:resource="&kno;Time-Span"/>
  <rdfs:comment>묘호, 일정한 기간을 가지고 있음
당 태종(627-649)</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#TempleNameYear -->
<owl:Class rdf:about="&kno;TempleNameYear">
  <rdfs:subClassOf rdf:resource="&kno;Time-Span"/>
  <rdfs:comment>묘호와 연도가 붙어서 연도를 나타내는 경우
太宗1(627)</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Term -->
<owl:Class rdf:about="&kno;Term">
  <rdfs:comment>고유명사, 개념용어를 제외한 일반적인 용어</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Time -->
<owl:Class rdf:about="&kno;Time"/>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Time-Span -->
<owl:Class rdf:about="&kno;Time-Span">
  <rdfs:subClassOf rdf:resource="&kno;Time"/>
  <rdfs:comment>역사적 시대나 날짜와 같이 특정 기간 (Time-Span)을 나타내기 위해 사용되는 모든 형태
의 이름 또는 코드로 그 정확도에 있어 차이가 있음</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Tribe -->
<owl:Class rdf:about="&kno;Tribe">
  <rdfs:subClassOf rdf:resource="&kno;Agent"/>
  <rdfs:comment>종족 </rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Works -->
<owl:Class rdf:about="&kno;Works">
  <rdfs:subClassOf rdf:resource="&kno;Artifacts"/>
  <rdfs:comment>저작, 작품</rdfs:comment>
</owl:Class>
```

```
<!-- http://dh.aks.ac.kr/ontologies/historicevent/kno#Year -->
<owl:Class rdf:about="&kno;Year">
  <rdfs:subClassOf rdf:resource="&kno;Time"/>
  <rdfs:comment>연도, 기원전, 기원후 포함 </rdfs:comment>
</owl:Class>
```

```
</rdf:RDF>
```

```
<!-- Generated by the OWL API (version 3.4.2) http://owlapi.sourceforge.net -->
```