Journal of the Korean Institute of Industrial Engineers
Vol. 34, No. 4, pp. 460-469, December 2008.

자취 군집화를 통한 프로세스 마이닝의 성능 개선

송민석[1] · C.W. Gűnther[1] · W.M.P. van der Aalst[1] · 정재윤[2†]

[1]아인트호벤공대 기술경영학부 / [2]경희대학교 산업공학과

Improving Process Mining with Trace Clustering

Minseok Song[1] · C.W. Gűnther[1] · W.M.P. van der Aalst[1] · Jae-Yoon Jung[2]

[1]Faculty of Technology Management, Eindhoven University of Technology, The Netherlands
[2]Department of Industrial Engineering, Kyung Hee University, Gyeonggi-Do, South Korea

Process mining aims at mining valuable information from process execution results (called “event logs”). Even though process mining techniques have proven to be a valuable tool, the mining results from real process logs are usually too complex to interpret. The main cause that leads to complex models is the diversity of process logs. To address this issue, this paper proposes a trace clustering approach that splits a process log into homogeneous subsets and applies existing process mining techniques to each subset. Based on log profiles from a process log, the approach uses existing clustering techniques to derive clusters. Our approach are implemented in ProM framework. To illustrate this, a real-life case study is also presented.

Keywords: Process Mining, Trace Clustering, Workflow, Data Mining, SOM

1. 서 론

수많은 기업을 비롯하여 공공 및 서비스 조직에서는 업무의 체계적인 수행과 관리를 위하여 비즈니스 프로세스를 이용하고 있다. 일반적으로 비즈니스 프로세스는 정보시스템을 통해 자동화되어 관리되고 있으며, 이러한 프로세스 처리가 가능한 정보시스템을 프로세스 인식 정보시스템(PAIS : Process-Aware Information Systems)라고 부른다(Dumas *et al*., 2005). PAIS는 비즈니스 프로세스를 정확하게 실행하고 효율적으로 관리하는 기능을 수행한다. PAIS에서는 특정 목적을 달성하기 위하여 비즈니스 프로세스가 실제로 실행되는 일련의 과정을 프로세스의 인스턴스(instance) 또는 케이스(case)라고 부르며, 이 케이스를 수행하는 과정에서 발생되는 업무 실행 기록들을 이벤트 로그(event log)로 기록하여 저장한다.

이처럼 비즈니스 프로세스 실행 과정에서 누적된 이벤트 로그를 분석함으로써 유용한 정보를 추출하는 것을 프로세스 마이닝(process mining)이라고 부른다(Aalst *et al*., 2007a). 프로세스 마이닝은 업무의 의존 관계를 표현할 수 있는 프로세스 모델을 유도하거나(Aalst *et al*., 2004), 업무의 상관관계, 작업자의 업무 전달 관계 등의 프로세스 수행상의 특징을 분석하거나(Aalst *et al*., 2005), 프로세스 실행이 정의된 모델에 따라 정확히 실행되는지에 관한 일치성 검사(conformance checking)를 수행하는 데(Rozinat and Aalst, 2008) 활용된다.

비즈니스 프로세스 자동화를 위해 전통적으로 활용되던 트랜잭션 워크플로우(transactional workflow)의 경우, 대부분 미리 기술된 프로세스 정의를 벗어나는 경우가 상대적으로 흔하지 않았다. 그러나 헬스케어, 제품개발, 고객지원 등의 실제 기업의 업무 흐름들은 매우 가변적이고 유연하게 운영되기 때문에 실제 업무 수행 결과를 분석하는 것이 점점 더 중요해지고 있다. 실행된 업무 이력을 분석하여 프로세스 지식을 발견하고 비교 분석하는 것은 프로세스의 지속적인 개선과 품질 향상에 있어서 필수적인 사항이며, 프로세스 마이닝은 이러한 목적으로 활용될 수 있다.

기존의 프로세스 마이닝 기법들을 근본적으로 유연하고 복

이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-352-D00208).
† 연락저자 : 정재윤, 446-701 경기도 용인시 기흥구 서천동 경희대학교 산업공학과, Tel : 031-201-2537, Fax : 031-205-4004,
E-mail : jyjung@khu.ac.kr
2008년 7월 2일 접수; 2008년 10월 11일 게재 확정.

잡한 환경에 적용하는 데에는 한계가 있다. 그 이유는 가변적이고 유연한 업무 환경에서는 다양한 잠재적 업무들이 존재하며, 업무들의 상호 의존관계가 비정형적이고 비구조적이다. 이로 인해, 실제 정보시스템에서 추출된 프로세스 모델들은 대부분 단위 업무의 개수가 지나치게 많거나, 업무간 관계가 매우 복잡하여 사람이 이해하거나 실행 가능한 프로세스 정의로 변환하기가 힘든 경우가 대부분이다.

본 연구에서는 이러한 프로세스 마이닝의 한계를 극복하고 성능을 향상시키기 위하여 계층적으로 프로세스 마이닝을 적용하기 위한 자취 군집화(trace clustering) 방법을 적용한다. 자취 군집화는 프로세스 실행에서 발생한 이벤트 로그를 분석하여 유형별로 발생한 프로세스 케이스들을 군집화하는 방법이다. <Figure 1>은 본 논문의 연구 방법을 도식화 하고 있다. 먼저 각 프로세스 케이스의 자취 프로파일(trace profile)을 생성하고, 자취 프로파일에 기반하여 여러 거리 척도로 상호 거리를 측정한다. 이 거리 척도는 이벤트 로그에 존재하는 두 케이스 간의 상대적 거리를 계산하는 데 사용된다. 그 다음 단계로, 클러스터링 알고리즘을 적용하여 서로 밀접한 관계에 있는 케이스들을 군집화하는데, 각 군집들은 유사한 목적을 수행하기 위한 케이스들로 그룹화되기 때문에 개별적으로 분석될 수 있고, 특히 유연한 환경에서 프로세스 마이닝 결과의 품질을 향상시킬 수 있다. 본 연구에서는 이러한 과정을 구현한 시스템을 활용하여, 실제 사례 연구에 적용함으로써 효과를 입증하였다.

본 논문은 다음과 같이 구성된다. 먼저 제 2절에서 관련 연구를 소개하고, 제 3절에서 본 논문의 이해를 돕기 위한 예제 프로세스를 설명한다. 제 4절과 제 5절에서 자취 프로파일과 자취 군집화 방법에 대해서 소개를 하고, 제 6절과 제 7절에서 구현 결과와 실제 사례 응용 결과를 소개한다. 마지막으로 제 8절에서 본 논문의 결론을 맺는다.

2. 관련 연구

프로세스 마이닝은 기업정보시스템에서 프로세스 실행과정에서 발생한 "이벤트 로그"로부터 의미있는 지식을 추출해내는 과정이다. 즉, 비즈니스 프로세스의 액티비티 실행 과정에서 누적된 기록을 통하여 비즈니스 프로세스의 개선이나 설계에 필요한 유용한 지식을 추출하는 것이다. 프로세스 저장소의 실행 결과 및 이벤트에 관한 로그를 추출하여 기존의 통계적 기법, 인공지능 알고리즘, 사회적 네트워크 기법 등을 통하여 분석함으로써(Medeiros *et al.*, 2007; Jansen-Vullers *et al.*, 2006; Rozinat and Aalst, 2006; Aalst *et al.*, 2005), 업무의 의존 관계를 표현할 수 있는 프로세스 모델을 유도하거나, 업무의 상관관계, 작업자의 업무 전달 관계 등의 프로세스 수행 상의 특징을 분석한다(Aalst and Basten, 2002).

최근 여러 가지 기법과 도구들이 개발되었는데, 대부분의 연구는 프로세스 로그로부터 프로세스 모델을 추출하는 것이었다. 본 연구에서는 아인트호벤 공대에서 개발한 오픈 프레임웍인 ProM 시스템의 플러그인(plug-in) 형태로 구현되었는데(Aalst *et al.*, 2007b), 이 플랫폼은 ERP나 BPM 시스템의 프로세스 로그를 임포트하여 프로세스 모델 추출, 프로세스 모델 분류, 프로세스 패턴 추출, 작업자간 업무전달 형태 등을 분석할 수 있는 환경을 제공한다. 본 연구에서 제시된 프로세스 로그 클러스터링 기법도 ProM 플랫폼 상에 적용함으로써 입력 데이터 포맷을 공유하고, 다른 알고리즘과 연계할 수 있도록 구현되었다.

여러 문헌에서 프로세스 마이닝의 적용가능성을 보여주고 있으며(Aalst *et al.*, 2007a), 나아가 일부 논문에서는 실제 프로세스 로그로부터 추출되는 프로세스 모델을 단순화하기 위한 접근들이 일부 있었다. Gűnther and van der Aalst(2007)는 프로세스 모델을 단순화하기 위한 퍼지 마이닝 기법을 고안하였

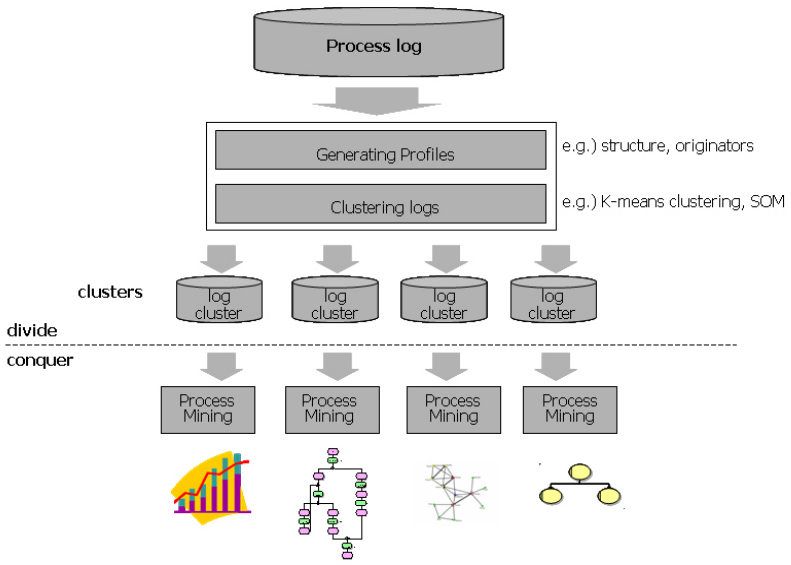

Figure 1. Methodology of Trace Clustering

고, Greco *et al.* (2006)은 프로세스 모델을 발견하기 위하여 로그를 분류하는 방법을 제시하였으며, Jung(2008)은 프로세스 로그를 클러스터링하기 위하여 단위작업과 연관관계들을 벡터로 표현하여 분류하는 방법을 적용하였다.

본 연구에서는 Greco *et al.* (2006)의 연구와는 달리 클러스터링에 필요한 속성들을 조절함으로써 사용자가 원하는 적절한 결과를 추출할 수 있으며, Jung(2008)을 비롯한 기존의 프로세스 로그 분석 기법들이 단지 활동의 이름과 활동 간의 선후관계만을 사용한 반면에, 본 논문에서는 활동 및 선후관계뿐만 아니라, 통제흐름, 조직구성, 데이터 등 여러 가지 측면들을 고려하여 프로세스 실행 결과를 표현할 수 있는 프로파일을 생성하였으며, 이를 이용하여 프로세스의 클러스터를 생성한다. 본 연구의 클러스터링을 위해 적용된 방법은 데이터 마이닝 분야에서 자주 활용되는 일반적인 기법들로서(Heyer *et al.*, 1999; Kaufman and Rousseeuw, 1990; Kohonen, 1982; Lloyd, 1982), 프로세스 마이닝 분야에 적용하여 어떻게 기존 기법들과 결합하여 활용할 수 있는지를 보여준다.

3. 예제 프로세스

본 장에서는 본 논문에서 다루고 있는 프로세스 케이스와 이벤트 로그의 이해를 돕기 위해 간단한 진료 프로세스를 예로 들어 설명한다. 진료 프로세스는 환자의 등록 (A), 기본적인 검사 (B), 보험 가입 사항을 확인 (C), 의사의 진단 및 치료 (D), 과거 진료 기록 확인 (E)으로 구성이 된다. <Table 1>은 진료를 받은 5명의 환자가 어떤 단계를 거쳤는지를 나타내고 있다. 표의 각 줄은 각 환자가 거친 단계를 나타내는데, 본 논문에서는 이를 하나의 프로세스 케이스라고 한다. 표는 총 18개의 이벤트를 시간 순으로 기록하였다. 예를 들어, Case1의 (A, John)은 Case1의 가장 첫 번째 이벤트로서, 첫 번째 환자에 대한 등록 업무를 John이 수행했다는 것을 알려준다.

<Table 1>에 기록된 이벤트 로그에 프로세스 구조 마이닝 기법 중 하나인 α-알고리즘[3]을 사용하면 <Figure 2>의 왼쪽과 같은 페트리넷 모델을 얻게 된다. 모든 케이스들이 환자 등록 과정 (A)으로 시작되고, 상담 및 치료 과정 (D)로 끝나기 때문에 작업 A와 작업 B가 가장 앞과 뒤에 오게 된다. 중간에는 기본적인 검사를 하는 단계 (B)와 보험가입사항을 확인하는 단계 (C)가 병렬로 진행되는 경우와 과거 진료 기록 확인 (E) 경우가 있다. 즉 작업 B와 작업 C가 동시에 일어나거나 작업 E가 일어나게 된다. 작업 B와 작업 C가 일어나는 경우는 병원을 처음 방문하는 경우이고, 병원을 재방문한 경우에는 작업 E가 일어나게 된다. 따라서 <Figure 2>와 같은 모델을 얻게 된다.

Table 1. Event log of medical treatment process

Case ID	Event logs
1	(A, John), (B, Sue), (C, John), (D, Pete)
2	(A, John), (C, Mike), (B, John), (D, Sue)
3	(A, Carol), (E, Mike), (D, Sue)
4	(A, Pete), (C, Carol), (B, Clare), (D, Pete)
5	(A, Sue), (E, Pete), (D, Clare)

프로세스를 업무의 종류에 따라서 구분하여 {Case1, Case2, Case4}와 {Case3, Case5}의 두 개의 그룹으로 나누어 α-알고리즘을 적용하면, <Figure 2>의 오른편과 같은 두 개의 보다 단순한 모델을 얻을 수 있다. 즉, 군집화를 통해서 프로세스 로그를 비슷한 성질을 가지고 있는 케이스들로 분류하고, 여기에 기존의 프로세스 마이닝 기법을 적용하면 보다 이해하기 쉬운 프로세스 마이닝 결과를 얻을 수 있다.

4. 자취 프로파일

군집화 알고리즘을 적용시키는 데 있어서, 군집화의 대상이 되는 데이터를 정의하는 것이 매우 중요하다. 본 장에서는 프로세스 로그와 여기서 추출할 수 있는 데이터에 대해서 설명한다.

4.1 프로세스 로그

프로세스 로그는 프로세스의 수행 정보를 담고 있다. 본 논문에서는 일반적으로 프로세스 마이닝에서 많이 쓰이고 있는 MXML(Mining XML) 포맷으로 정의된 프로세스 로그를 사용한다. 업무를 지원하는 정보 시스템은 공통적으로 작업과 작업의 수행자 등에 대한 정보를 제공하는데, MXML 이런 일반적인 업무 시스템에서 기록이 되는 프로세스 수행 관련 기록

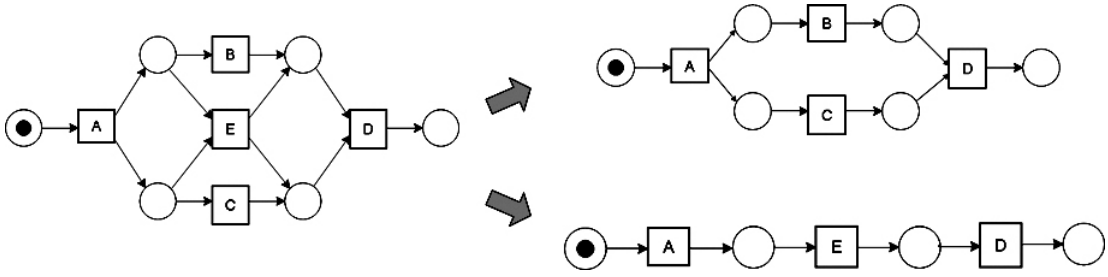

Figure 2. The result of mining the event log in Table 1

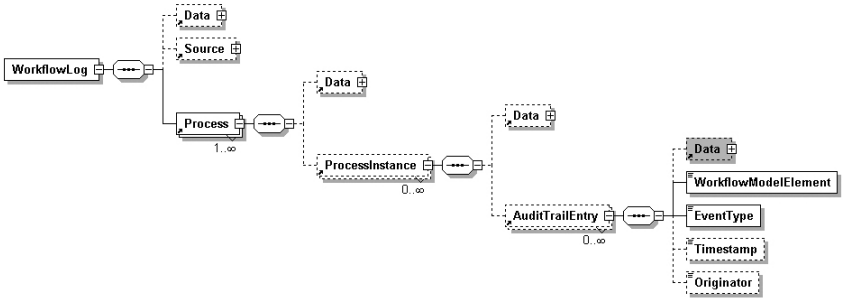

Figure 3. Schema of MXML format

을 저장할 수 있는 표준 포맷으로, 프로세스와 관련된 다양한 정보를 담을 수 있다. <Figure 3>은 MXML의 구조를 도식화 하여 나타낸 그림이다. MXML은 *WorkflowLog*를 최상위에 가지고 있다. *WorkflowLog*는 *Data*와 *Source*를 갖고 있는데, 이를 통해서 프로세스 로그를 생성한 시스템과 프로세스 로그에 관련된 메타 정보를 저장할 수 있다. *WorkflowLog*는 여러 개의 *Process*가 포함이 될 수 있고, 각 *Process*는 여러 *ProcessInstance*를 갖는다. 하나의 *ProcessInstance*는 하나의 케이스를 담게 된다. *ProcessInstance*는 여러 개의 *AuditTrailEntry*를 갖는다. 각 *AuditTrailEntry*는 하나의 이벤트를 나타내게 되는데, 예를 들어 특정 업무의 시작과 종료 등을 기록하게 된다. *AuditTrailEntry*에는 *WorkflowModelElement*, *EventType*, *Timestamp*, *Originator*를 하위에 갖게 된다. *WorkflowModelElement*는 작업의 이름을 나타내며, *EventType*은 시작, 보류, 종료 등의 이벤트의 종류를 가지고 있다. *Timestamp*는 그 이벤트가 발생한 시각을 나타내고, *Originator*는 이벤트를 수행한 수행자를 나타낸다.

4.2 로그 프로파일

앞장에서 설명한 MXML 포맷의 프로세스 로그는 다양한 종류의 정보를 가지고 있다. 군집화를 위해서 여러 관점의 데이터를 추출할 수 있는데, 이런 다양한 관점의 정보를 하나의 프로파일(profile) 로 정의한다. 프로파일은 기본적으로 작업(activity) 프로파일, 수행자(Originator) 프로파일, 전이(Transition) 프로파일, 데이터(Data) 프로파일, 성능(Performance) 프로파일 등이 있다. <Table 2>는 각 프로파일에 대한 설명을 담고 있다.

각 프로파일들은 <Table 3>과 같이 하나의 통합된 표의 형태로 표현이 될 수 있다. <Table 3>은 <Table 2>의 프로세스 로그에서 추출한 로그 프로파일의 예를 나타내는데, 작업 프로 파일과 전이 프로파일을 보여주고 있다. 예를 들어 Case1의 경우 작업 A, B, C, D가 한 번씩 수행이 되었기 때문에, 각 항목이 1의 값을 갖는다. 또한 작업 A, 작업 B, 작업 C, 작업 D의 순서로 수행이 되었기 때문에, AB, BC, CD 전이 항목이 1의 값을 갖게 된다.

Table 2. Profile generated from event log

작업 프로파일	각 케이스에서 수행이 되었던 작업에 대한 정보로 작업이 수행되었던 수로 표시
수행자 프로파일	각 케이스의 수행에 참여한 수행자의 정보로 각 수행자가 케이스의 수행에 참여한 수로 표시
전이 프로파일	자취에 나타난 전이 정보로 두 작업의 쌍으로 표시되며, 두 작업이 전이 관계에 있었던 빈도로 표시. 예를 들어 작업 A가 수행된 후에 작업 B가 수행된 경우에, (A, B)쌍으로 표시된다.
데이터 프로파일	각 케이스가 가지고 있는 데이터 값으로 생성이 되는 프로파일
성과 프로파일	작업의 수행성과에 대한 프로파일로 각 작업의 최소, 평균, 최대 수행 시간 등과 각 케이스의 수행시간 등의 값을 갖는다.

Table 3. An example profile extracted from event log in <Table 1>

case ID	Activity Profile					Transition Profile								…
	A	B	C	D	E	AB	AC	AE	BC	BD	CB	CD	ED	…
1	1	1	1	1	0	1	0	0	1	0	0	1	0	…
2	1	1	1	1	0	0	1	0	0	0	1	1	0	…
3	1	0	0	1	1	0	0	1	0	0	0	0	1	…
4	1	1	1	1	0	0	1	0	0	0	1	1	0	…
5	1	0	0	1	1	0	0	1	0	0	0	0	1	…

프로파일들은 프로세스 마이닝의 목적에 따라서 선택 가능하며, 중요도에 따라서 가중치를 부여할 수도 있다. 예를 들어 프로세스 구조에 따라서 프로세스 로그를 분리할 경우에는 수행자나 업무의 성과에 대한 프로파일은 생략하고, 업무 프로파일과 전이 프로파일만 선택할 수 있다. 또한 이들의 가중치를 부여함으로써 업무의 수행 빈도에 더 초점을 맞추거나 업무의 전이에 더 초점을 맞출 수도 있다.

5. 군집화 방법

군집화는 기본적으로 비슷한 값을 가지는 변수를 하나의 군집으로 묶는 것이다. 이때 변수 사이의 유사성을 계산하는 거리 계산 방법과 군집화 알고리즘이 군집화 결과에 중요한 영향을 끼치게 된다. 본 논문에서는 프로세스 케이스 사이의 거리를 계산하는 방법으로 유클리디안(Euclidian) 거리, 해밍(Hamming) 거리, 자카르드(Jaccard) 거리를 사용한다. 앞장에서 설명한 케이스의 프로파일은 n차원의 벡터로 표현이 될 수 있다(여기서 n은 로그에서 추출한 아이템의 수). 즉 케이스 c_j는 $<i_{j1}, i_{j2}, \cdots, i_{jn}>$로 표현이 되고, i_{jk}는 케이스 j의 k번째 아이템의 값을 나타낸다. 여기서 각 거리는 다음과 같이 정의된다.

- 유클리디안값 = $\sqrt{\sum_{l=1}^{n}|i_{jl}-i_{kl}|^2}$
- 해밍값 = $\sum_{l=0}^{n}\delta(i_{jl}, i_{kl})/n$,
  $\delta(i_{jl}, i_{kl}) = \begin{cases}0, & if(i_{jl}>0 \wedge i_{kl}>0) \vee (i_{jl}=i_{kl}=0)\\ 1, & otherwise\end{cases}$
- 자카르드값 = $1-(\sum_{l=0}^{n}i_{jl}*i_{kl})/(\sum_{l=1}^{n}i_{jl}^2+\sum_{l=1}^{n}i_{kl}^2-\sum_{l=1}^{n}i_{jl}i_{kl})$

유클리디안값은 일반적으로 가장 많이 사용이 되는 거리계수이다. 하지만, 변수들의 분산이 매우 큰 경우에는 좋은 결과를 나타내지 못한다. 해밍값은 변수가 0의 값을 갖는지 아니면 0보다 큰 값을 갖는지만 고려하기 때문에 변수들의 분산이 큰 경우에 유용하게 쓰일 수 있다. 자카르드값은 군집화 연구 분야에서 많이 쓰이는 방법의 하나로 0에서 1사이의 값을 갖는다.

이 세 가지 거리 계수는 다양한 군집화 알고리즘에 사용할 수 있다. 본 논문에서는 군집화 알고리즘으로 K-평균 군집, QT (Quality Threshold) 군집, 계층적 군집(AHC : Agglomerative Hierarchical Clustering), SOM(Self-Organizing Map) 방법을 사용한다.

- **K-평균 군집화** : K-평균 군집화는 일반적으로 가장 많이 사용이 되고 있는 군집화 방법으로 데이터를 k개의 군집으로 분리하는 방법이다. K-평균 군집화 방법은 다음과 같이 정의된다.

정의 1. K-평균 군집화.

(i) Initialize 무작위로 k개의 중심값 $\mu_1, \mu_2, \cdots, \mu_k$를 초기화 시킴
(ii) Do 각 케이스와 μ_i의 거리를 계산하여 가장 가까운 군집에 할당
(iii) 각 군집에 할당된 케이스를 바탕으로 새로운 μ_i 계산
(iii) Until μ_i값들의 변화가 없을 때까지 반복
(v) Return k개의 군집 반환

K-평균 방법은 알고리즘의 속도가 빠르다는 장점이 있지만, 그 결과가 초기값에 의존적이기 때문에 적당한 군집의 개수와 초기 중심값을 설정하는 것이 중요하고, 다수의 실험을 통해 적당한 군집을 얻는 것이 중요하다.

- **QT 군집화** : QT 알고리즘은 생명공학 분야에서 유전자의 분류를 위해서 발명된 알고리즘이다. QT는 앞서 설명한 K-평균 방법보다 수행 시간이 오래 걸리지만, 임계값을 정해주면, 항상 동일한 군집 결과를 얻을 수 있고, 초기 군집의 개수를 정할 필요가 없다. 보통 QT 방법은 균형 잡히지 않은 결과 값을 갖게 되는데, 하나의 큰 군집과 상대적으로 작은 여러 개의 군집을 생성한다. 따라서 큰 군집을 바탕으로 데이터의 일반적인 특징을 파악하는데 유용하게 쓰일 수 있다. 즉 프로세스의 주요 흐름 등을 파악하는 데 유용하게 쓰일 수 있다.
- **계층적 군집화(Agglomerative Hierarchial Clustering)** : K-평균 군집화와 QT 방법과 상이하게 계층적 군집화는 비슷한 케이스를 묶어 나감으로써 군집을 구성한다. 즉 n개의 케이스가 있다고 가정하면, 각 케이스를 바탕으로 초기에 n개의 군집을 생성하고, 가까운 거리에 위치한 군집들을 묶어 나감으로써 군집을 생성한다. 군집의 결과는 보통 덴드로그램(dendrogram)으로 표시가 되는데, 케이스와 군집들 사이의 거리 정보를 보여주기 때문에 사용자가 적당한 군집을 선택하는 데 도움이 된다.
- **SOM 방법(Self-Organizing Map)** : SOM 방법은 신경망 기법의 하나로 다차원의 데이터를 저차원(주로 2차원)의 공간에 사상 시켜 군집을 형성한다. 즉 비슷한 설질을 갖는 데이터를 저차원 공간의 비슷한 공간에 위치시킴으로서 같은 공간이나 인접한 공간에 위치한 데이터는 하나의 군집에 속하게 된다. 군집은 훈련 단계와 할당 단계를 통해서 이루어진다. 훈련 단계를 통해서 군집이 형성되는 저차원 공간을 초기화 하고, 할당 단계를 통해서 각 케이스들을 저차원 공간에 할당하게 된다. 군집화의 결과는 사각형이나 육각형 모양의 셀로 표시가 되며, 매우 빠르게 수행이 된다는 장점이 있다.

6. 시스템 구현

본 장에서는 시스템 구현에 대해서 설명한다. 시스템은 ProM

프레임웍에 구현이 되었다. ProM 프레임웍은 프로세스 마이닝 기법의 구현을 위한 기반 프레임웍으로 프로세스 로그를 다루는 데 필요한 필수 기능들이 제공되기 때문에 이를 재사용 할 수 있고, 프로세스 마이닝 방법을 플러그인의 형태로 손쉽게 추가할 수 있다. <Figure 4>는 ProM 프레임웍의 주요 기능을 나타낸다. 로그 필터는 로그를 읽어 필요한 부분을 추출해 내는 역할을 하고, 읽혀진 로그는 마이닝 플러그인을 통해서 분석될 수 있다. 마이닝 플러그인의 결과는 분석 플러그인을 통해서 분석할 수 있고, 변환 플러그인을 통해서는 모델간의 변화가 가능하다. 예를 들어 페트리넷 모델을 EPC 모델이나 YAWL 모델 등으로 변화 시키는 것이 가능하다. 또한 결과를 저장하고 다시 프레임웍에 로딩하는 것이 가능하다.

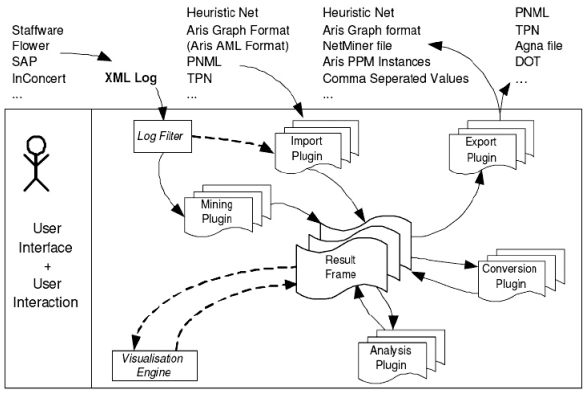

Figure 4. ProM Framework

본 논문에서 제시한 자취 군집화 방법은 분석 플러그인의 형태로 개발이 되었다. 프로세스 로그에 군집화 알고리즘을 적용하여 다양한 부분 로그들을 생성하고, 이 로그들은 ProM 프레임웍에서 지원하는 다양한 프로세스 마이닝 방법을 통해서 분석할 수 있다. <Figure 5>는 구현 결과를 나타낸다. 그림의 왼쪽 부분에 프로세스 로그에서 생성할 수 있는 프로파일 리스트를 보여준다. 사용자는 관련 프로파일을 선택하고, 선택된 프로파일에 대하여 가중치를 부여할 수 있다. 오른쪽 부분에서는 유사도 측정 방법과 알고리즘을 선택할 수 있다. 본 논문에서 다루고 있지 않았지만, 데이터 마이닝 기법에서 많이 쓰이는 데이터 전처리를 설정할 수 있다.

설정이 끝나면, 선택된 군집화 알고리즘에 따라서 군집화 화면으로 넘어간다. 예를 들어 <Figure 6>은 QT 방법을 선택한 후에 나오는 화면이다. 이 화면을 통해서 각 알고리즘에 따른 세부 설정이 가능하다. <Figure 6>의 경우, QT 알고리즘에 쓰이는 임계값을 오른쪽에 부분에서 선택할 수 있고, 왼쪽에는 각 군집과 군집에 속한 케이스들의 유사도 관계를 색상으로 보여주고 있다.

7. 사례 연구

본 연구 결과는 여러 기관에서 얻은 다양한 프로세스 로그의 분석에 사용이 되었다. 예를 들어 네덜란드의 시청, 병원 등의 구매 프로세스와 필립스의 소프트웨어 테스트 프로세스 분석 등에 사용이 되었다. 본 논문에서는 이들 사례 중 하나인 네덜란드 암스테르담 대학 병원의 프로세스 로그 분석 결과를 소개 한다. 프로세스는 산부인과 관련 프로세스로 질병을 진단하는 부분과 치료하는 부분을 포함하고 있다. 프로세스 로그에는 2005년부터 2006년까지의 진료 기록을 담고 있는데, 총 619명의 환자의 진료 기록을 포함하고 있다. 병원의 프로세스는 각 환자가 하나의 프로세스 케이스로 분류가 되기 때문에 619명의 환자는 619개의 프로세스 케이스로 기록이 되어 있다. 단위 작업으로 총 52개의 진단 작업이 있고, 34개의 부서가 작업의 수행에 참여하고 있다. 참고로 진료 작업은 본 프로세스 포함이 되어 있지 않다. 프로세스 로그에는 총 3,574개의 이벤

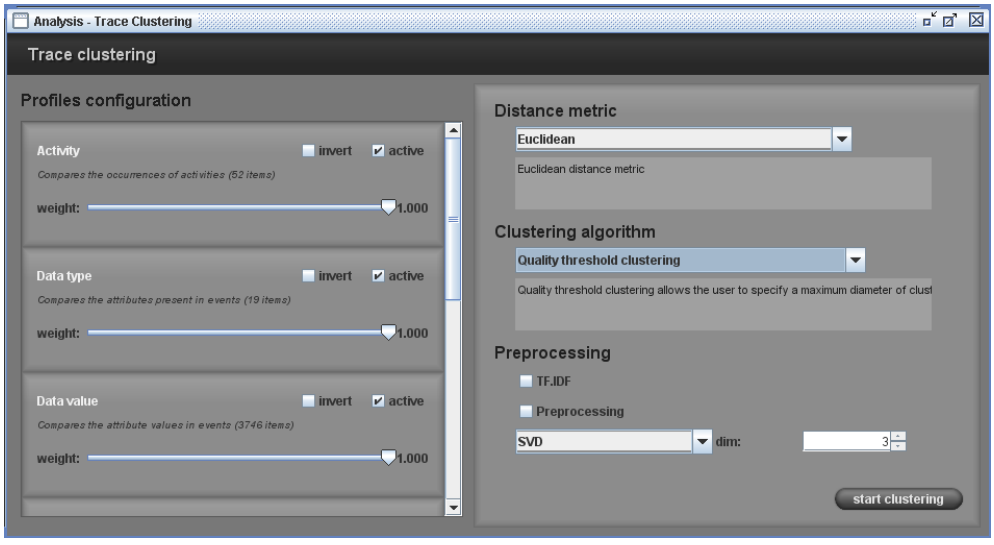

Figure 5. Setting for trace clustering

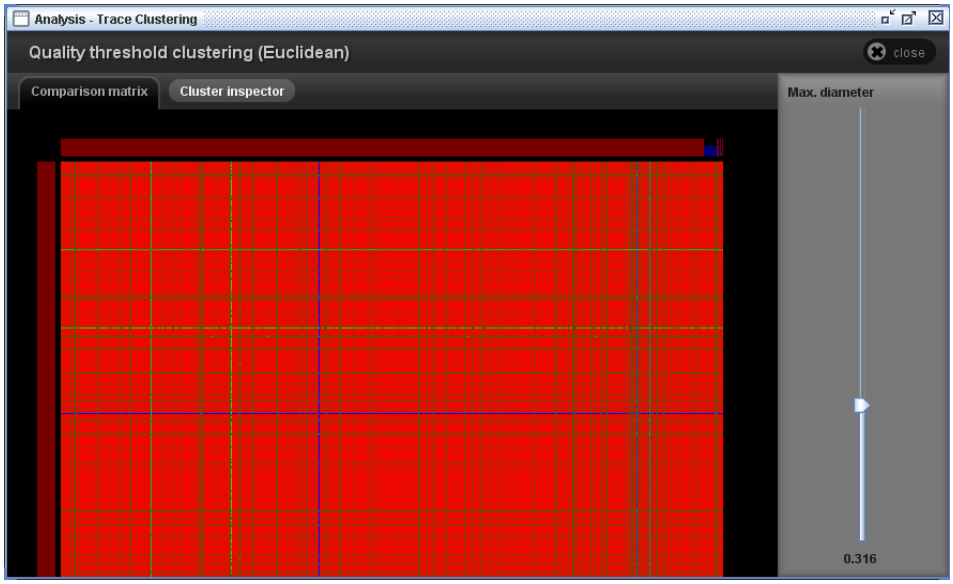

Figure 6. Result of trace clustering

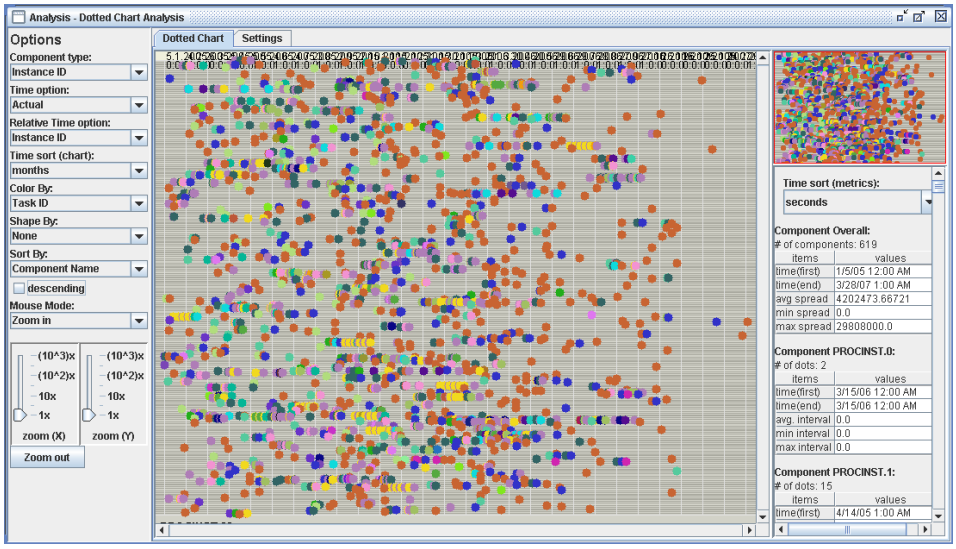

Figure 7. Scattering of event logs

트가 기록되어 있는데, 각 이벤트는 단위 작업의 종료 이벤트를 나타낸다. <Figure 7>은 이벤트의 분포를 보여 주고 있다. 그림에서 각 원은 로그에 있는 이벤트를 나타내며, 원의 색상은 이벤트가 나타내는 작업을 나타낸다. 세로축은 케이스를 나타내고, 가로축은 시간을 나타내며, 각 이벤트들은 관련 케이스와 발생 시간에 따라서 정렬이 되어 있다. 그림에서 보이듯이 프로세스 로그에 많은 이벤트들이 복잡하게 나타나고 있음을 알 수 있다.

위의 로그에서 프로세스 모델을 추출하였다. <Figure 8>은 전체 프로세스 로그에서 프로세스 모델을 도출한 결과이다. 본 논문에서는 휴리스틱 마이닝 기법을 사용하였다. 휴리스틱 마이닝 기법은 노이즈에 둔감하고, 빠르게 수행이 되기 때문에 프로세스 모델 마이닝에 많이 사용이 된다. 그림의 모델은 휴리스틱넷의 형태로 프로세스를 나타내고 있는데, 사각형은 단위 작업을 나타내고, 화살표는 작업들 사이의 선후 관계를 나타낸다. 그림에서 보이듯이 추출된 모델은 52개의 진단 작업이 매우 복잡하게 얽혀 있음을 알 수 있다. 이 모델의 이해를 위하여 업무를 담당하는 담당의사에게 보여주었지만, 프로세스의 흐름을 파악할 수 없었다.

모델을 보다 세분화하기 위해서 본 논문에서 제시한 군집화 방법을 적용하였다. 앞에서 제시한 유사도 측정 방법들과 군집화 방법들을 적용시킨 결과 유클리디안값과 SOM 방법을 적용했을 때보다 정확한 군집화 결과가 도출이 되었다. <Figure 9>는 SOM 방법의 적용 결과를 보여 주고 있다. 그림에서 각 셀은 하나의 군집을 나타내고, 각 점은 하나의 케이스를 나타낸다. 참고로 모델을 세분화 하는 것이 목적이기 때문에, 작업

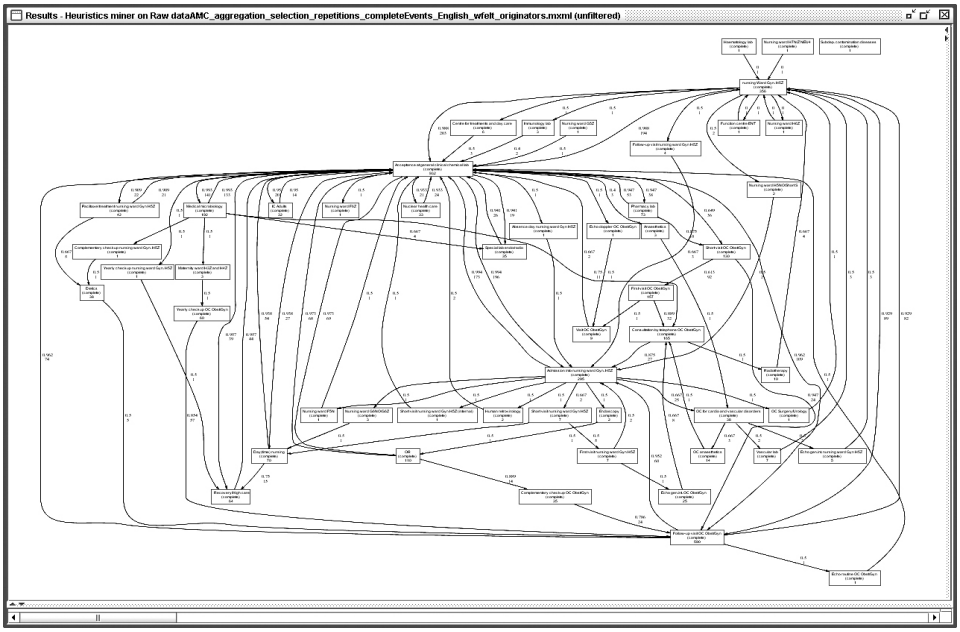

Figure 8. Process model extracted from a whole process log

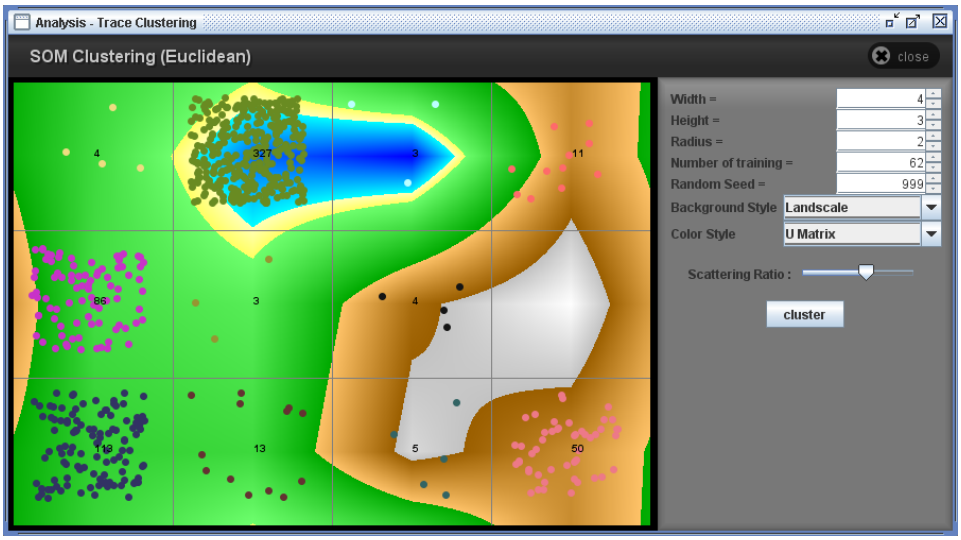

Figure 9. Result of SOM analysis

프로파일만을 사용하였다. 클러스터링의 결과 11개의 군집을 얻었다. 이 중에서 321개의 케이스가 속한 셀(1, 2)와 113개의 케이스가 속한 셀(3, 1)에서 휴리스틱 마이닝 기법으로 프로세스 모델을 도출하였다.

<Figure 10>의 왼쪽 모델은 셀(1, 2)에서 도출한 프로세스 모델이다. 전체 케이스의 절반이 넘는 352개의 케이스를 사용하였지만, 도출된 프로세스 모델은 단지 11개의 단위 작업으로 구성이 되어있고, 이전 모델에 비해서 상대적으로 매우 간단한 모델이 도출되었다. <Figure 10>의 오른쪽 모델은 셀(2, 0)에서 도출된 모델이다. 이 모델은 113개의 케이스에서 추출이 되었지만, 이전 모델과 비슷하게 매우 복잡한 형태를 가지고 있다. 이 두 개의 프로세스를 프로세스의 담당의사와 함께 분석하였다. 분석 결과 셀(1, 2)에서 도출한 모델은 지역 병원에서 진단을 받고, 치료를 받기 위해 종합병원인 암스테르담 병원을 찾은 환자들이 일반적으로 따르는 프로세스임을 알 수 있었다. 이들은 해당 부서를 찾아 간단한 진단을 받고, 치료를 위해 다른 부서로 이관이 되기 때문에 상대적으로 간단한 프로세스를 따르게 된다. 셀(2, 0)에서 도출된 모델은 지역 병원을 방문하지 않고 바로 암스테르담 병원을 찾은 환자들이 진단을 받는 일반적인 프로세스를 나타내고 있다. 이 경우에는 보다 복잡하고 세밀한 진단 프로세스를 따름을 알 수 있다.

본 사례 연구에서 알 수 있듯이 본 논문에서 제시한 군집화 방법을 통해서 효과적으로 프로세스 로그를 군집화 할 수 있음을 알 수 있었고, 이를 통해서 보다 이해하기 쉬운 프로세스

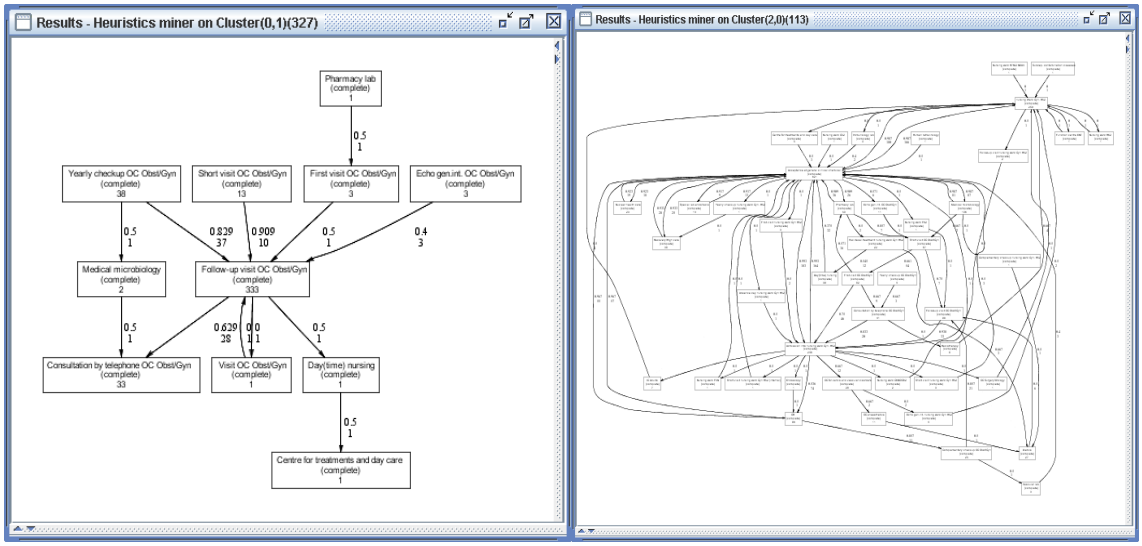

Figure 10. Process model extracted from a process cluster

마이닝 결과를 얻을 수 있음을 알 수 있다.

8. 결 론

프로세스 마이닝은 프로세스의 업무 수행 결과를 분석함으로써 실제 업무가 어떻게 이루어지는지를 분석할 수 있는 기법을 제공한다. 특히 기업의 업무 프로세스가 정형화 유연한 업무 환경의 프로세스인 경우에 프로세스 마이닝 결과는 실제 업무가 어떻게 수행이 되는지에 대한 유용한 정보를 추출할 수 있다. 하지만, 실제 프로세스 실행 로그를 이해하고 획득하는 것이 간단하지 않으며, 기업의 업무는 매우 복잡하기 때문에 분석 결과가 매우 복잡하고, 그 결과에서 기업의 업무를 파악하는 것이 쉽지 않다는 문제가 있었다.

이런 문제를 해결하기 위해서 본 연구에서는 군집화 방법을 적용시킴으로써 업무 수행 결과를 비슷한 성질을 가진 여러 개의 군집으로 분리하여, 단위 군집을 분석함으로써 보다 이해 가능한 프로세스 마이닝 결과를 도출하는 방법을 제안하였다. 프로세스 로그에서 프로세스의 성질을 나타내는 프로파일에 대해서 정의하고, 여러 가지 군집화 알고리즘을 적용시키는 방법에 대해서 설명하였다. 본 논문에서 제안한 방법은 ProM 프레임웍 상에 하나의 플러그인으로 구현이 되었으며, 구현 결과를 여러 프로세스 로그에 적용하여 그 활용 가능성을 검증하고 그 중 한 가지 사례 연구를 소개하였다.

향후 연구 주제로는 보다 정확한 군집 결과를 얻기 위한 연구가 필요하다. 여러 사례 연구를 통해서 K-군집 방법과 SOM 방법이 프로세스 로그를 비교적 잘 군집화 하는 경향이 있다는 관찰을 하게 되었다. 하지만 보다 과학적으로 거리 측정 방법들과 군집화 알고리즘들이 프로세스의 특성과 어떤 연관성을 가지고 있는지에 대한 연구가 필요하다. 이를 통해서 사용자들이 프로세스의 특성에 따라서 보다 적합한 군집화 알고리즘을 선택하는데 도움이 될 것이다. 또한 군집화 결과를 설명하는 연구가 필요하다. 어떤 요소들이 서로 다른 군집을 형성하는데 영향을 끼쳤는지에 대한 연구가 필요하다.

참고문헌

van der Aalst, W. M. P. and Basten, T. (2002), Inheritance of workflows : an approach to tackling problems related to change, *Theoretical Computer Science*, **270**(1), 125-203.

van der Aalst, W. M. P., Weijters, A. J. M. M., and Maruster, L. (2004), Workow Mining : Discovering Process Models from Event Logs, *IEEE Transactions on Knowledge and Data Engineering*, **16** (9), 1128-1142.

van der Aalst, W. M. P., Reijers, H. A., and Song, M. (2005), Discovering Social Networks from Event Logs, *Computer Supported Cooperative work*, **14**(6), 549-593.

van der Aalst, W. M. P., H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. Alves de Medeiros, M. Song, and H. M. W. Verbeek (2007), Business Process Mining : An Industrial Application, Information Systems, *Information Systems*, **32**(5), 713-732.

van der Aalst, W. M. P., et al. (2007), ProM 4.0 : Comprehensive Support for Real Process Analysis, *Proc. 28th Int'l Conf. on Applications and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, *Lecture Notes on Computer Science*, **4546**, 484-494.

Dumas, M., van der Aalst, W. M. P., and ter Hofstede, A. H. M. (2005), *Process-Aware Information Systems: Bridging People and Software through Process Technology,* Wiley and Sons.

Greco, G., Guzzo, A., and Pontieri, L. (2006), Discovering Expressive Process Models by Clustering Log Traces, *IEEE Transactions on Knowledge and Data Engineering*, **18**(8), 1010-1027.

Gűnther, C. W. and van der Aalst, W. M. P. (2007), Fuzzy Mining -Adaptive Process Simplication Based on Multi-Perspective Metrics, In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management(BPM 2007)*,

Lecture Notes on Computer Science, **4714**, 328-343.

Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999), Exploring Expression Data: Identification and Analysis of Coexpressed Genes, *Genome Research*, **9**(11), 1106-1115.

Jansen-Vullers, M. H., van der Aalst, W. M. P., and Rosemann, M. (2006), Mining Configurable Enterprise Information Systems, *Data and Knowledge Engineering*, **56**(3), 195-244.

Jung, J.-Y., PROCL : A Process Log Clustering System, *The Journal of Society for e-Business Studies*, **13**(2), 181-194.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data* : *An Introduction to Cluster Analysis*.

Kohonen, T. (1982), Self-organation formation of topologically correct feature maps, *Biological Cybernetics*, **43**(1), 59-69.

Lloyd, S. P. (1982), Least squares quantization in PCM, *IEEE Transactions on Information Theory*, **2**, 129-137.

de Medeiros, A. K. Alves, Weijters, A. J. M. M., and van der Aalst, W. M. P. (2007), Genetic Process Mining : An Experimental Evaluation, *Data Mining and Knowledge Discovery*, **14**(2), 245-304.

Rozinat, A. and W. M. P. van der Aalst (2006), Decision Mining in ProM, *Proc. 4th Int. Conf. on Business Process Management*, 420 -425.

Rozinat, A. and van der Aalst, W. M. P. (2008), Conformance checking of processes based on monitoring real behavior, *Information Systems*, **33**(1), 64-95.