

A study on resource-constrained composite web service design

민영빈*, 김동수**, 강석호*

*서울대학교 산업공학과, **송실대학교 산업정보시스템공학과

A study on resource-constrained composite web service design

Min, Yeongbin, Kim, Dongsoo, Kang, Suk-ho

Seoul National University, Soongsil University

E-mail : bluesboybin@naver.com, dskim@ssu.ac.kr

요약

기업들 간의 경쟁이 치열해짐에 따라 기업들은 최종 제품뿐만 아니라 최종 제품을 생산하는 프로세스에 대한 효율에 관심을 갖기 시작하였다. 또한 기업들은 자사의 핵심 역량을 중심으로 기업 구조를 재편하고, 핵심 역량 이외의 부분은 아웃소싱을 맡기고 있다. 이러한 상황에서 기업은 다양한 파트너들과 비즈니스 프로세스를 공유하고, 결합 관계에 따라 유연하게 구성할 수 있어야 한다. 기업의 비즈니스 프로세스에 대한 중요성을 일찍 깨닫고 많은 기업들이 프로세스 혁신 작업을 수행하였다. 몇몇 기업은 주목할만한 큰 성공을 거두었음에도 불구하고 기대했던 성과를 거두지 못한 채 많은 비용만 소모한 기업도 존재한다. 이와 같은 일들은 비즈니스 프로세스 디자인을 위한 방법에 대한 개선이 필요함을 시사한다. 하지만 대부분의 방법은 프로세스 디자인을 시각화하고, 평가하는데 중점을 둔 분석 도구의 기능에 편중되어 있으며, 여전히 프로세스의 실제 디자인을 지원하는 형식적인 모델링에 대한 연구는 여전히 부족한 실정이다. 따라서 본 논문에서는 비즈니스 프로세스의 모델링의 응용 분야로 기업 어플리케이션 통합으로 각광받고 있는 웹 서비스를 선택, 웹 서비스로 이루어진 프로세스의 최적화된 디자인을 지원하는 수리 모델을 제시하고자 한다.

1. 서론

기업들 간의 경쟁이 치열해짐에 따라 기업들은 최종 제품뿐만 아니라 최종 제품을 생산하는 프로세스에 대한 효율에 관심을 갖기 시작하였다. 더욱이 기업들은 자사의 핵심 역량을 중심으로 기업 구조를 재편하고, 핵심 역량 이외의 부분은 아웃소싱을 맡기는 추세이고, 이러한 상황에서 기업은 다양한 파트너들과 비즈니스 프로세스를 공유하고, 결합 관계에 따라 유연하게 구성할 수 있어야 한다. 이와 같은 변화는 프로세스가 기업의 경쟁력을 결정짓는 요소로 자리하게 되었음을 의미하며, 급변하는 경영 환경에 대응하여 기업은 자신의 비즈니스

프로세스의 구조를 빠르게 대응할 수 있어야 시장에서 성공을 누릴 수 있게 된 것이다.

그리고 기업 구조가 핵심 역량을 중심으로 재편되는 가정에서 기업 간 협업이 중요해지고, 이 협업을 위한 어플리케이션으로 웹 서비스가 각광을 받고 있다. 과거 기업 내부에서 프로세스의 혁신을 추구하던 것과 달리 웹 환경에서는 기업의 비즈니스 프로세스는 기업 외부와 연결되어 다양한 형태로 구현될 수 있다. 기업은 UDDI에 등록된 서비스 정보를 토대로 보다 느슨하게 다른 기업과 협업 체계를 구성할 수 있으며, 이를 바탕으로 변화에 빠르게 대응할 수 있는 보다 유연한 프로세스 구

조를 가질 수 있다.

이처럼 최근 많은 기업들은 비즈니스 프로세스에 대한 중요성을 깨닫고 프로세스 혁신 작업을 수행하였다. 몇몇 기업은 주목할만한 큰 성공을 거두었지만, 기대했던 성과를 거두지 못한 채 많은 비용만 소모한 기업도 대다수 존재한다 [2]. 이는 다른 회사의 프로세스를 벤치마킹하는 것과 새로운 기술을 도입하는 것이 확실한 개선을 가져오는 것이 아님을 알 수 있으며, 비즈니스 프로세스 혁신 활동에도 방법론이 필요함을 시사한다. 지금까지 비즈니스 프로세스 리엔지니어링을 위한 몇 가지 개념적인 프레임워크가 개발되어왔다 [3]. 하지만 대부분의 방법은 프로세스 디자인을 시각화하고, 평가하는데 중점을 둔 분석 알고리즘 및 도구의 개발에 편중되어 있으며 [4], 프로세스의 디자인을 지원하는 형식적인 모델링에 대한 연구는 여전히 부족한 실정이다.

게다가 웹 서비스들로 이루어진 플로우와 관련된 대부분의 기존 연구들은 서비스들의 단계와 상태전이 관계, 즉 프로세스 모델을 정의한 상태에서 적절한 컴포넌트 서비스를 찾는 데 초점을 맞추고 있다. 하지만 프로세스 모델의 각 스텝을 미리 정의할 수 없는 상황이거나, 초기 디자인을 작성해야 하는 경우에는 여전히 전문가에게 의존하고 있는 실정이다. 웹 환경에서 웹 서비스는 다양한 형태로 제공되기에 미리 정의된 프로세스의 여러 단계를 포함하는 복잡한 서비스가 있을 경우, 최적의 파트너 선정이 이루어지지 않을 수 있다. 따라서 본 논문에서는 웹 서비스로 이루어진 프로세스의 최적화된 디자인을 지원하는 수리 모델을 제시해보도록 하겠다.

논문의 구성은 다음과 같다. 2장에서는 웹 서비스 디자인과 밀접한 Hofacker의 프로세스 디자인 모델을 소개한다. 그 다음 3장에서는 Hofacker의 논문을 확장하여 웹 환경에서의 서비스 디자인 작성을 도와주는 분석 모델을 제시한다. 4장에서는 새로 제시한 모델을 토대로 한 실험 설계와 그 결과를 보여주고, 마지막으로 5장에서 결론과 추후 연구 방향에 대해 다루도록 하겠다.

본 연구는 서울시 산학연 협력사업의 연구비지원에 의하여 연구되었음

2. 관련 연구

웹 서비스로 이루어진 플로우는 여러 웹 서비스들이 모여 하나의 프로세스를 이루는 것으로 볼 수 있기에 프로세스 디자인과 깊은 관련을 갖는다. Hofacker는 비즈니스 프로세스 디자인의 최적화에 대한 알고리즘을 제시하였다 [5]. Hofacker는 이 논문을 통해 비즈니스 프로세스 디자인 문제에 대한 분석 방법을 고안하였다.

2.1. 프로세스 디자인 모델

Hofacker는 프로세스를 표현하는 일반적인 프레임워크를 개발하였다. 이 프레임워크에서 프로세스를 표현하는 주된 객체는 액티비티와 리소스이다. Hofacker는 리소스를 물리 리소스와 정보 리소스로 구분하였고, 액티비티는 이 리소스들을 입력과 출력으로 사용하는 전이 단계로 정의하였다. Hofacker는 비즈니스 프로세스 모델을 정의하기에 앞서 다음과 같은 가정을 하였다.

Assumptions)

- i. 액티비티는 하나의 입력 리소스를 사용하여 하나의 출력 리소스를 생성한다.
- ii. 액티비티의 입력 리소스들은 OR 조건이 아닌 AND 조건으로 들어온다.
- iii. 액티비티의 출력 리소스들은 AND 조건으로 생성된다.
- iv. 생성된 물리 리소스는 하나의 액티비티에서만 입력으로 사용할 수 있다.
- v. 비용은 서비스를 수행할 때 발생한다.

Hofacker는 비즈니스 프로세스를 다음과 같은 가정과 형태로 모델링하였다.

Notation)

- 액티비티 집합 $A = \{a_1, a_2, \dots, a_n\}$
- 리소스 집합 $B = \{b_1, b_2, \dots, b_m\}$
- BI - 정보 리소스 집합
- BP - 물리 리소스 집합
- i번째 액티비티의 속성 $v_i = \{v_{i1}, v_{i2}, \dots, v_{ik}\}$
- 입력 집합 - $I_i \subseteq B$
- 출력 집합 - $O_i \subseteq B$
- 글로벌 입력 - I_{glob} , 글로벌 출력 - O_{glob}
- x_i - 액티비티 a_i 가 프로세스 디자인에 포함되어

있는지 여부를 가리키는 이진 변수

· y_j - 리소스 b_j 가 프로세스 내에서 사용되는지

여부를 가리키는 이진 변수

· 프로세스 디자인 $P = \{(a_i, p_i), (), \dots, ()\}$

· 액티비티 집합과 출력 리소스와의 관계를 나타내는 행렬 T 의 성분 t_{ij}

$$t_{ij}=1 \text{ if } b_j \in O_i$$

$$t_{ij}=0 \text{ otherwise}$$

· λ_{ij} - 액티비티 a_i 가 리소스 b_j 를 만드는데 사용되었는지 여부를 표현하는 이진 변수

· 액티비티 a_i 의 시작 시간 - p_i

· 리소스 b_j 가 사용가능해지는 시간 - q_j

$$\min \sum_{i,j} v_i x_i$$

$$x_i \leq r_{ij} \quad \forall i, j: b_j \in I_i, b_j \in B_p$$

$$x_i \leq y_j \quad \forall i, j: b_j \in I_i, b_j \in B_l$$

$$g_{o_j} + \sum_i r_{ij} \leq M \cdot g_{i_j} + \sum_i t_{ij} x_i \quad \forall j: b_j \in B_p$$

$$y_j \leq g_{i_j} + \sum_i t_{ij} x_i \quad \forall j: b_j \in B_l$$

$$y_j \geq g_{o_j}$$

$$p_i \geq q_j - M(1 - x_i) \quad \forall i, j: b_j \in I_i$$

$$q_j \leq p_i + \delta_i + M(1 - x_i) \quad \forall j: b_j \in O_i$$

$$q_j \geq p_i + \delta_i - M(1 - x_i) - M(1 - \lambda_{ij}) \quad \forall i: b_j \in O_i$$

$$\lambda_{ij} \leq x_i \quad \forall i, j: b_j \in O_i$$

$$\sum_{i: b_j \in O_i} \lambda_{ij} \geq \sum_i r_{ij} + g_{o_j} - M(1 - y_j) \quad \forall j: b_j \in B_p, g_{i_j} = 0$$

$$\sum_{i: b_j \in O_i} \lambda_{ij} \geq 1 - M(1 - y_j) \quad \forall j: b_j \in B_p, g_{i_j} = 0$$

$$x_i \in \{0,1\}$$

$$y_j \in \{0,1\}$$

$$\lambda_{ij} \in \{0,1\} \quad \forall i, j: b_j \in O_i$$

2.2. 한계

Hofacker는 앞에 제시한 모델을 정수계획법, 분지한계법, 유전 알고리즘으로 그 성능을 실험하였고, 세 기법 모두 합리적인 시간 내에서 결과값을 획득할 수 있었다. 하지만 Hofacker가 제시한 비즈니스 프로세스 모델은 몇 가지 한계를 가진다.

우선 액티비티가 만들어내는 출력 리소스에 유연성이 결여되어 있다. Hofacker가 제시한 모델에서 어떤 액티비티는 실행 후에 항상 같은 리소스를 생성한다. 하지만 실제 비즈니스 프로세스 내의 액

티비티는 입력된 요소에 따라 다른 출력값을 생성할 수도 있다. 대출 심사의 예를 들면 고객의 신용 등급에 따라 대출을 승인하는 메시지가 출력될 수도 있고, 다른 신용 정보를 조회하라는 메시지가 출력될 수도 있다.

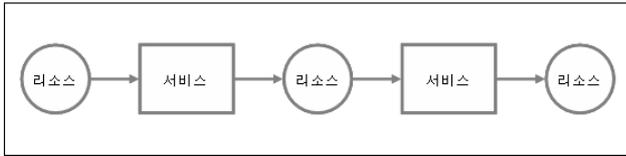
그리고 OR, AND, XOR 분기 등으로 연결되어 있는 복잡하게 구성된 액티비티들을 처리하는데 한계가 있다. 예를 들어 고객의 위험도를 평가하는 액티비티가 있고, 이 액티비티는 세 곳 이상에서 평가한 자료를 기반으로 최종 결론을 내린다고 하자. 이러한 경우 Hofacker 모델에서는 고객의 위험도를 평가하기 위한 입력 리소스들을 모두 획득한 후에 결론을 내려야만 한다. 기업 내에서 이루어지는 액티비티들로 프로세스를 디자인할 경우에 이는 큰 문제가 아니지만, 웹과 같이 거대한 공간에서 적절한 웹 서비스들을 검색하여 프로세스를 디자인할 경우, 많은 노력과 비용이 필요하다.

따라서 본 논문은 액티비티에 해당하는 서비스의 출력물에 보다 유연성을 부여하면서, 복잡하게 구조화된 서비스를 고려한 프로세스 디자인 모델을 제시하고자 한다.

3. 문제 정의

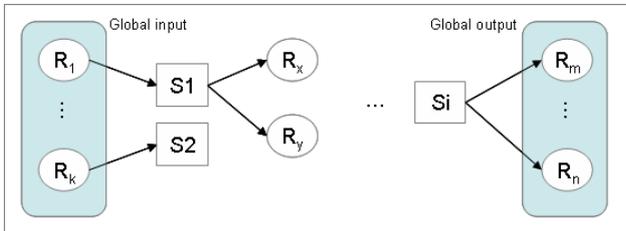
본 논문에서 다루는 웹 서비스 디자인 문제는 특정 리소스들을 만들기 위해 필요한 서비스들의 스케줄링 문제라고 볼 수 있다. 웹 서비스 디자인에서 관심 객체는 서비스와 리소스이다. 일반적으로 하나의 웹 서비스는 하나 이상의 리소스를 입력으로 받고, 이를 사용하여 하나 이상의 결과물 즉 출력 리소스를 만들어낸다. 웹 서비스 디자인 문제에서 특정 리소스는 입력과 출력의 형태로 어떤 서비스와 관계를 가지게 되고, 이는 확정적이다. 일반적으로 웹 서비스를 사용하기 위해서는 WSDL에 정의된 입력과 출력을 따르게 되는데, 이 입력과 출력으로 들어올 수 있는 대상을 리소스라 볼 경우, 서비스와 리소스의 관계는 고정되어 있기 때문이다.

서비스, 리소스, 그리고 이들 사이의 관계는 유방향 그래프(directed graph)의 형태로 표현 가능하다. 그래프에서 각각의 서비스와 리소스는 노드로 표현된다. Hofacker가 제시한 모형과 마찬가지로 각각의 서비스들은 리소스를 매개로 연결이 된다.



<그림 1> 서비스와 리소스의 연결 형태

각 서비스는 실행에 필요한 리소스와 실행 후 만들어지는 리소스가 정해져 있고, 서비스를 수행하지 않고도 활용할 수 있는 자원인 글로벌 리소스가 존재한다. 목표로 하는 최종 산출물을 생성하기 위해 글로벌 리소스를 기반으로 서비스를 실행시키고, 이 서비스들의 결과물을 중간 입력 리소스로 활용하여 궁극적으로 최종 산출물을 얻는다.



<그림 2> 유방향 그래프로 표현된 웹 서비스 디자인

각각의 서비스는 지속시간, 수행비용, 품질 등과 같은 속성을 가진다. 이 속성들은 해당 서비스 내에 있는 오퍼레이션이 수행될 때 비용을 부가한다. 웹 서비스 디자인은 주어진 입력 요소를 활용하여 최종 결과물을 최소 비용으로 생성하는 것을 목적으로 한다. 즉 글로벌 리소스, 서비스 셋, 중간 단계의 리소스, 최종 산출물을 가지고 유방향 그래프를 작성하는데, 이 그래프의 서브그래프 중에서 모든 최종 산출물에 대해 어떤 글로벌 리소스가 존재하여 이 글로벌 리소스에서 최종 산출물로 가는 경로가 존재하는 그래프가 이 문제의 가능해이다. 그리고 이 가능해 중에서 비용을 최소화 하는 서브그래프를 찾는 것이 목적이다.

본 논문에서 제시하는 수리 모델은 Hofacker가 제시한 모델을 발전시킨 것으로 개선된 내용은 다음과 같다.

- 입력 리소스에 따라 프로세스의 출력값이 달라질 수 있다.
- 서비스에 들어오는 입력 리소스의 개수를 제한할 수 있다.
- 서비스의 오퍼레이션에 따라 다른 비용이 발생한다.

이와 같은 요소들은 좀 더 일반적인 비즈니스 프로세스를 반영하였다. Hofacker의 모델에서 액티비티는 출력 리소스를 모두 생성해내지만, 실제 프로세스에서는 다양한 출력물을 생성할 수 있다. 따라서 어떤 서비스는 하나 이상의 오퍼레이션을 가질 수 있고, 이 오퍼레이션들은 각각 다른 리소스를 입력으로 사용할 수 있도록 모델링하였다. 그리고 서비스 단위가 아닌 오퍼레이션 단위로 비용이 발생하도록 하였다.

Assumptions)

- i. 서비스는 종류 당 하나의 입력 리소스를 사용하여 종류 당 하나의 출력 리소스를 생성한다.
- ii. 비용은 서비스를 수행하여 출력 리소스를 생성할 때 발생한다.
- iii. 생성되는 출력 리소스에 따라 다른 비용이 부여된다.
- iv. 글로벌 입력 리소스는 일부만 사용 가능하다.
- v. 최종 출력 리소스는 서비스들을 실행하여 만들어지는 결과물로 모두 생성해야 한다.

본 논문에서 제시하는 웹 서비스 디자인을 지원하는 수리 모델은 다음과 같다.

Notation)

- $G(V, E)$ - 유방향 그래프
- S - 서비스 집합, $S \subseteq V$
- R : 리소스 집합, $R \subseteq V$
- GR_i - 글로벌 입력 리소스 집합, $GR_i \subseteq V$
- GR_o - 최종 출력 리소스 집합, $GR_o \subseteq V$
- x_i - 노드 i 가 프로세스 디자인에 포함되어 있는지 여부를 가리키는 이진 변수, $i \in V$
- y_{ij} - 노드 i 에서 노드 j 로 가는 호가 선택되었는지 여부를 가리키는 이진 변수, $y_{ij} \in E$
- c_{ij} - 서비스 i 에서 리소스 j 를 생성할 때 발생하

는 비용

$\delta(i)$: 노드 i 로부터 나가는 호를 가지는 노드 집합

$$\begin{aligned} \min \quad & \sum_{i \in V} c_{ij} y_{ji} \\ x_i = 1, \quad & i \in GR_o \\ \sum_{i \in V} y_{ij} \geq x_j, \quad & \forall j \in V \\ \sum_{j \in \delta(i)} y_{ij} \leq (\# \text{of outgoing arcs from } i) \times x_i \\ \sum_{i \in GR_i} x_i \geq 1 \\ \sum_{i \in V} y_{ij} \geq k \cdot x_j, \quad & \forall j \in V \\ \sum_{j \in \delta(i)} y_{ij} \leq 1, \quad & \forall i \in R_p \\ \sum_{i \in \gamma(GRo)} y_{ij} \geq |GR_o| \\ y_{ij} \leq y_{jk} \\ x_i \in \{0, 1\} \\ y_{ij} \in \{0, 1\} \end{aligned} \quad \begin{matrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \\ (7) \\ (8) \end{matrix}$$

이 수리 모델의 식들이 지니는 의미는 [표 1]과 같다.

[표 1] 수리 모델의 설명

목적함수	서비스의 오퍼레이션 레벨에서 선택된 호에 발생된 비용의 합을 최소화
제약식(1)	최종 출력 리소스에 해당하는 노드는 반드시 선택됨
제약식(2)	어떤 노드 j 가 선택이 되면, j 로 들어오는 노드 중에서 최소 하나는 선택됨
제약식(3)	노드 i 에서 나가는 호 중 하나 이상 선택되면 노드 i 도 반드시 선택된다
제약식(4)	최종 입력 리소스는 하나 이상 선택되어야 함
제약식(5)	입력으로 들어가는 리소스의 개수와 만들어지는 리소스의 개수는 하나임
제약식(6)	특정 노드로 들어오는 호는 최소 k 개가 되어야 함
제약식(7)	본 문제의 구조에서 도출된 valid inequality

제약식(8)	서비스의 오퍼레이션에 정의된 입력 리소스와 출력 리소스들의 관계 정의
--------	--

4. 실험

실험을 위하여 가상의 서비스와 리소스를 생성하였다. 일반적인 웹이라는 공간에 존재하는 서비스와 리소스는 실제 무한히 많지만, 특정 도메인 내에 존재하는 리소스와 서비스는 한정적일 것이라는 가정 하에 실험을 설계하였다. 즉 UDDI에 저장된 특정 카테고리에 포함된 서비스만을 대상으로 다루고 있다는 가정 하에 실험을 수행하였다. 또한 실제 웹 환경에서 같은 대상을 다른 이름으로 지칭할 수도 있지만, 여기서는 그러한 변동을 고려하지 않았다. 이러한 부분을 해결하기 위해서는 리소스에 대한 시맨틱 검색과 같은 전처리가 필요하다. 하지만 이 부분은 본 논문에서 주로 다루고자 하는 주대상이 아니므로, 전처리가 되었다고 가정하였다.

웹 서비스들을 호출하고, 이들을 구성하기 위하여 서비스와 리소스를 정의하였다. 그리고 각각의 서비스는 1~4개의 입력 리소스와 1~2개의 출력 리소스를 가진다. 이 때 특정 서비스에서 같은 리소스를 입력과 출력으로 동시에 사용하는 경우는 배제하였고, 루프가 발생하는 구간은 사전에 체크를 하여 제거하였다.

실험은 서비스와 리소스의 개수를 변화를 주면서 최적 웹 서비스 디자인을 찾는데 소요되는 시간을 검사하였으며, 각 실험 조건으로 100회의 반복을 수행하였다. 수리 모델은 LINDO Systems사에서 개발한 프로그램을 이용하였으며, 실험에 사용된 컴퓨터의 사양은 AMD Athlon(tm) 64 Processor 3700+ 2.21 GHz, 1.00GB RAM이다. [표 2]는 실험 결과이다.

5. 결론

본 논문은 웹 환경에서 웹 서비스를 호출하여 프로세스를 구축해야 할 때 이를 지원하는 수리 모델을 제시하였다. 이 수리 모델은 Hofacker가 제안한 모델을 확장한 것으로 보다 일반적인 비즈니스

[표 3] 최적 웹 서비스 디자인을 구하는데 소요된 시간 (단위: 초)

		100 서비스	200 서비스	300 서비스	400 서비스	500 서비스
50 리소스	Mean	52.030	194.135	327.892	592.387	743.893
	S.D.	182.667	312.990	172.470	125.286	389.996
100 리소스	Mean	172.389	278.158	519.721	790.073	1327.218
	S.D.	117.207	239.641	278.934	445.611	492.334

프로세스를 표현하고 있으며, 서비스 내의 오퍼레이션에 따른 출력값 변화를 구조화할 수 있다. 입력값과 출력값이 명시되어 있는 웹 서비스 환경에 적합할 것으로 판단된다.

추후에는 모델의 성능을 개선할 수 있는 알고리즘 개발에 착수할 계획이다. 본 논문의 실험에서는 최대 500개의 서비스와 100개의 리소스를 가정하였지만, 특정 도메인에 국한되었다고 하더라도 서비스와 리소스는 이 이상일 것이고, 얼마든지 확장이 가능할 것이다. 본 논문의 주 목적이 웹 서비스 디자인을 지원하는 수리 모델을 제시하는 것이기에 알고리즘 개선을 위한 특별한 노력은 기울이지 않았으나 tabu search나 simulated annealing 등의 알고리즘을 사용할 경우 성능의 개선이 가능하다. 따라서 이후 연구는 제시한 모델이 실제 웹 환경에서 활용하기 위해 계산 성능을 향상시키는 알고리즘을 개발하고, 생성되는 수량을 컨트롤할 수 있는 정교한 모델을 개발할 예정이다.

[5] Hofacker I, Vetschera R, Algorithmical approaches to business process design, Computers & Operation Research, 2001; 28:1253-1275

[6] Vergidis K, Tiwari A, Majeed B, Roy R, Optimization of business design: An algorithmical approach with multiple objectives, International Journal of Production of Economics, 2007; 109:105-121

[참고문헌]

[1] Hammer M, Champy J. Reengineering the corporation. New York: Harper and Row, 1993
 [2] Remenyi D, Heafield A. Businss process re-engineering: some aspects of how to evaluate and manage the risk exposure, International Journal of Project Management 1996; 14:349-57
 [3] Hess T Brecht L. State of the Art des Business Process Redesign. Wiesbaden: Gabler, 1995
 [4] Bradley P, Browne J, Jackson S, Jagdev H. Business process re-engineering(BPR) - A study of the software tools currently available. Computers in Industry 1995;25:309-30