UC Santa Barbara
Computer Science Department

Object-aware Identification of Microservices

Mohammad Javad Amiri

COMPUTER SCIENCE
UC SANTA BARBARA

Microservices Architecture

- Forms a software system as a group of fine-grained, cohesive, and loosely coupled services
- Each service implements a small business capability
- Three principles of the architecture:
  - **Bounded Context:** related functionalities are combined into a single business capability
  - **Size:** each service provides only a single business capability (preserving granularity)
  - **Independence:** each service is operationally independent of others (loosely coupled and highly cohesive)

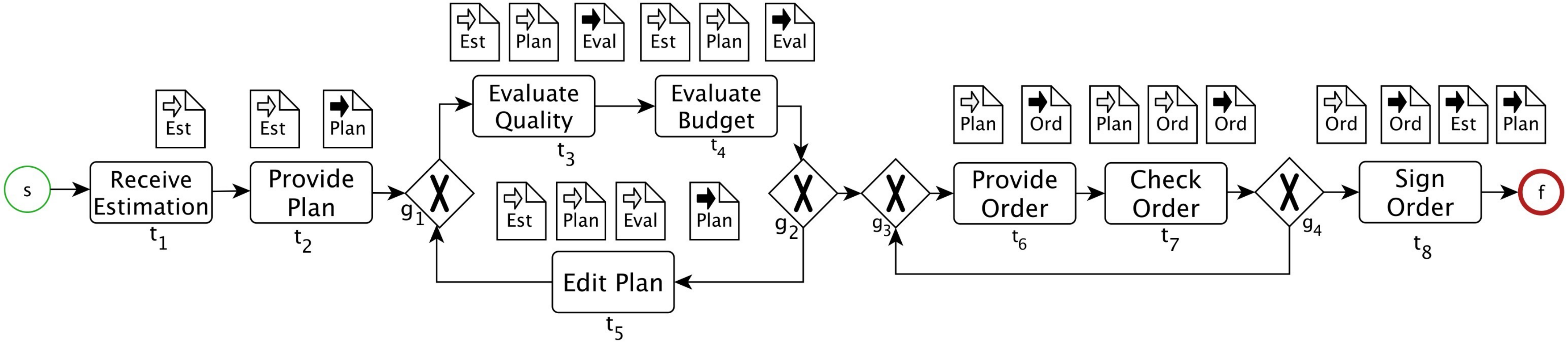
Microservices Identification

- Partitioning the system into microservices is usually performed intuitively
- If the functionalities of a system are highly interconnected, it is a challenging task
- We need to consider the underlying business functionalities of the system

Idea: identify microservices from business processes

Business Process

- A *business process* consists of a set of activities performed in coordination in an organizational environment to accomplish a business goal.
- A business process is defined as a tuple $P = (N, s, f, F, O, \rho, \omega)$.
- A *Plan Approval* process in a part supplier company



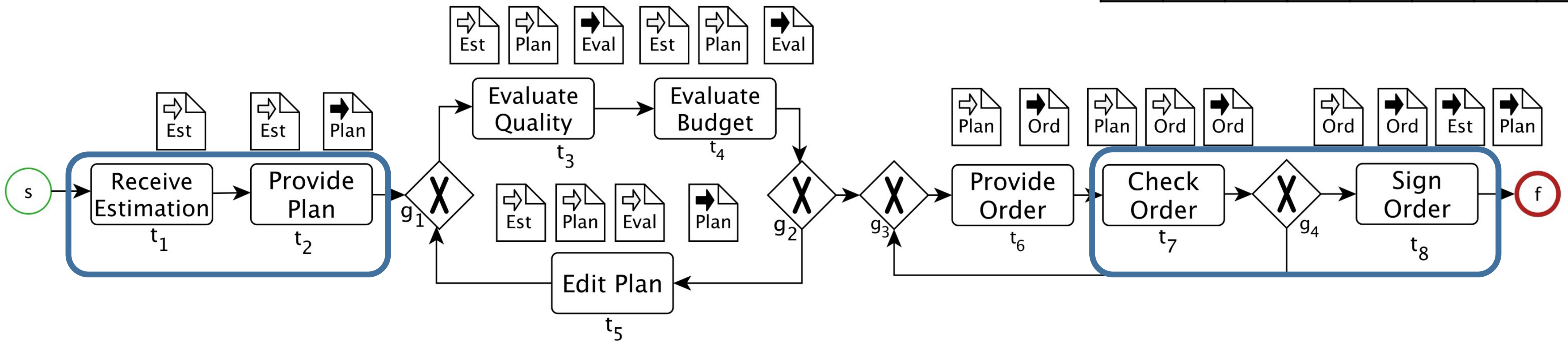
Microservice Identification Problem

- Decompose business processes into fine-grained, cohesive, and loosely coupled components which are called microservices.
- Each activity in a business process plays the role of an operation in a microservice.
- Given a process schema P, identify a set of services $\{S_1, S_2, \ldots, S_n\}$ where each service S_i has a set of distinct operations $\{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ such that s_{ij} is an activity node in P and if $s_1 = t$ and $s_2 = t$ then $s_1 = s_2$.

Structural Dependency of Activities ($T_P(a_i, a_j)$)

Given a schema $P = (N, s, f, F, O, \rho, \omega)$, for each pair of Activities $a_i, a_j \in N$, if $(a_i, a_j) \in F$ or there is a path $(a_i, n_1, \ldots, n_p, a_j)$ in P such that $\forall k \in [1 \ldots p]$: n_k is a gateway, then $T_P(a_i, a_j) = 1$, else $T_P(a_i, a_j) = 0$.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
t_1	0	1	0	0	0	0	0	0
t_2	0	0	1	0	0	0	0	0
t_3	0	0	0	1	0	0	0	0
t_4	0	0	0	0	1	1	0	0
t_5	0	0	1	0	0	0	0	0
t_6	0	0	0	0	0	0	1	0
t_7	0	0	0	0	0	1	0	1
t_8	0	0	0	0	0	0	0	0

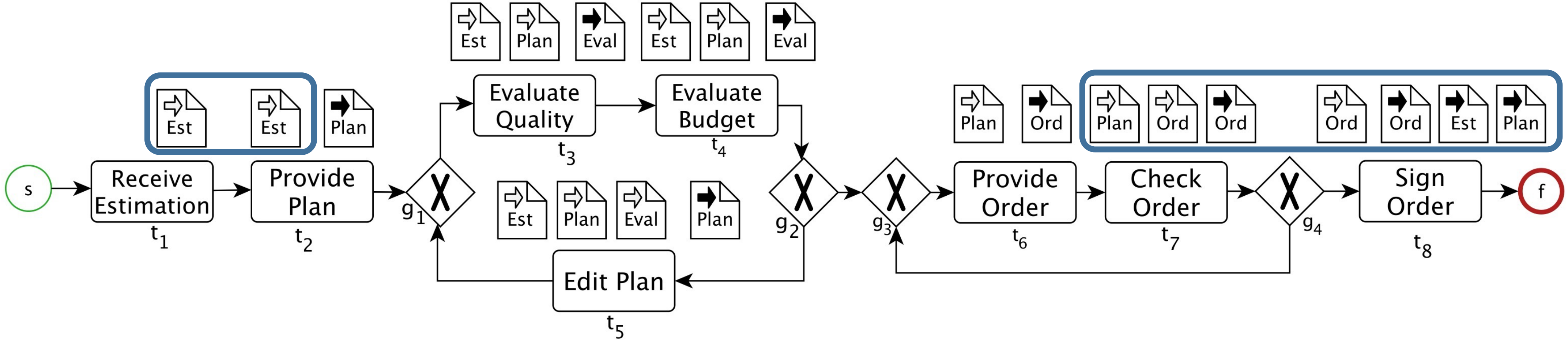


Object Dependency of Activities ($T_D(a_i, a_j)$)

Given a schema $P = (N, s, f, F, O, \rho, \omega)$, for each pair of Activities $a_i, a_j \in N$,

$$T_D(a_i,a_j) = |\omega(a_i) \cap \omega(a_j)| + 0.5 * |(\rho(a_i) \cap \omega(a_j)) \cup (\omega(a_i) \cap \rho(a_j))| + 0.25 * |\rho(a_i) \cap \rho(a_j)|$$

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
t_1	0	0.25	0.25	0.25	0.25	0	0	0.5
t_2	0.25	0	0.75	0.75	1.25	0.5	0.5	1.5
t_3	0.25	0.75	0	1.5	1.25	0.25	0.25	1
t_4	0.25	0.75	1.5	0	1.25	0.25	0.25	1
t_5	0.25	1.25	1.25	1.25	0	0.5	0.5	1.5
t_6	0	0.5	0.25	0.25	0.5	0	1.25	1.5
t_7	0	0.5	0.25	0.25	0.5	1.25	0	1.5
t_8	0.5	1.5	1	1	1.5	1.5	1.5	0

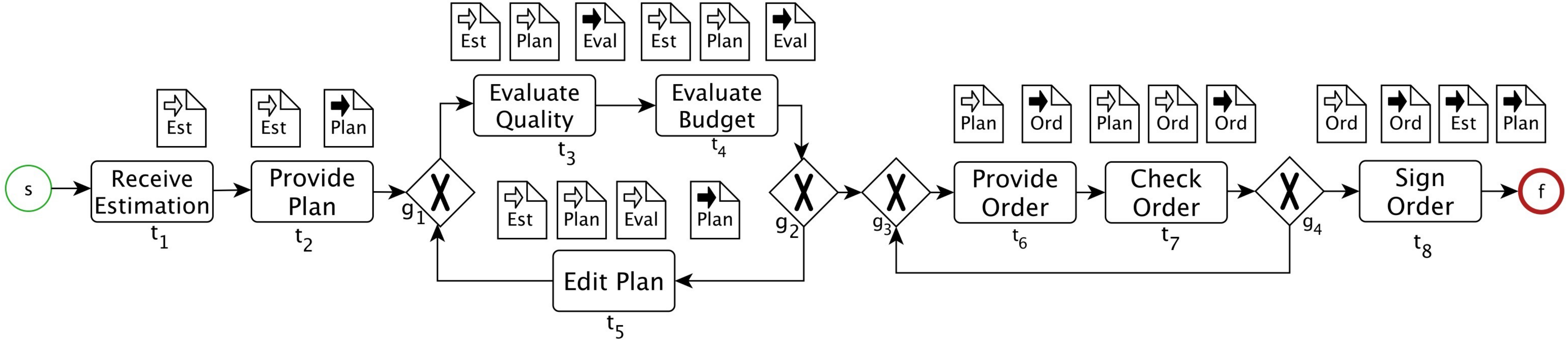


Total Dependency of Activities ($\Pi(a_i, a_j)$)

Given a schema $P = (N, s, f, F, O, \rho, \omega)$, for each pair of Activities $a_i, a_j \in N$,

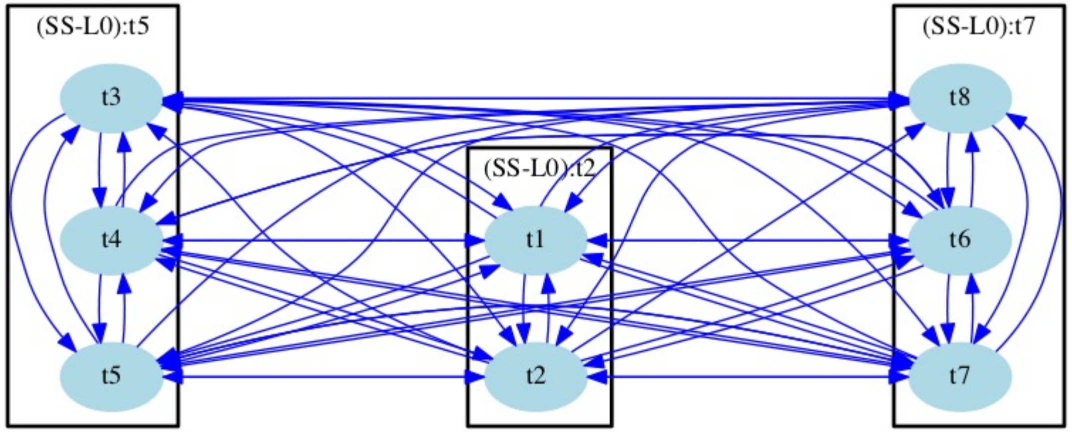
$$T(a_i,a_j) = T_P(a_i,a_j) + T_D(a_i,a_j)$$

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
t_1	0	1.25	0.25	0.25	0.25	0	0	0.5
t_2	0.25	0	1.75	0.75	1.25	0.5	0.5	1.5
t_3	0.25	0.75	0	2.5	1.25	0.25	0.25	1
t_4	0.25	0.75	1.5	0	2.25	1.25	0.25	1
t_5	0.25	1.25	2.25	1.25	0	0.5	0.5	1.5
t_6	0	0.5	0.25	0.25	0.5	0	2.25	1.5
t_7	0	0.5	0.25	0.25	0.5	2.25	0	2.5
t_8	0.5	1.5	1	1	1.5	1.5	1.5	0



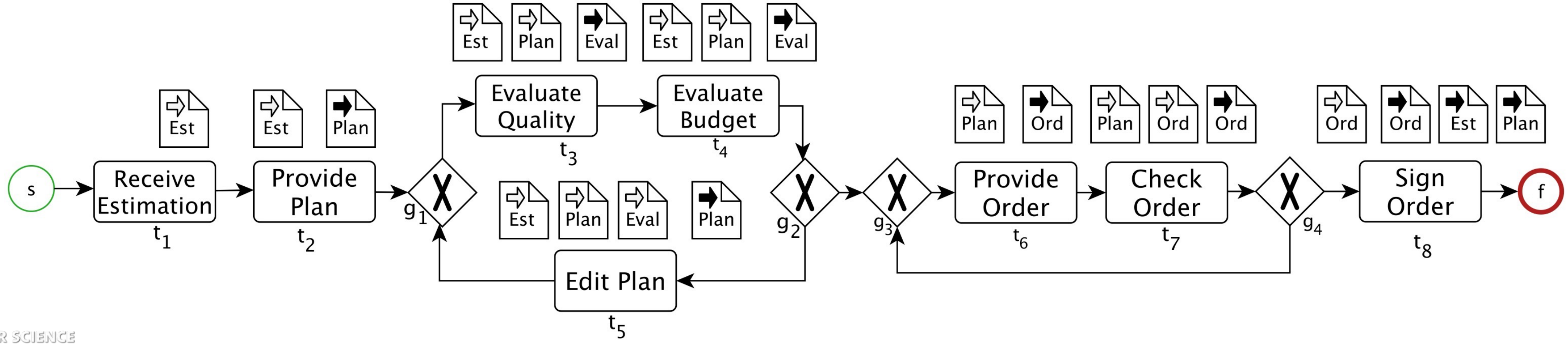
Microservice Identification

The final relation is clustered using a genetic algorithm and turbo-MQ fitness function.

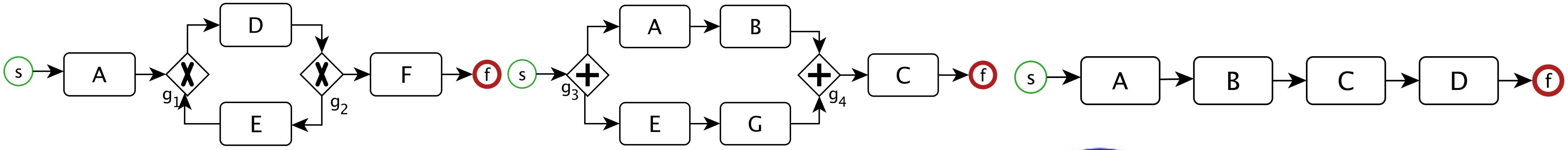


Turbo-MQ: $\sum_{i=1}^{k} CF_i$

$$\mathrm{CF}_i = \begin{cases} 0, & \mu = 0 \\ \dfrac{2\mu_i}{2\mu_i + \sum_{i=1, i \neq j}^{k} (\delta_{ij} + \delta_{ji})} & otherwise \end{cases}$$

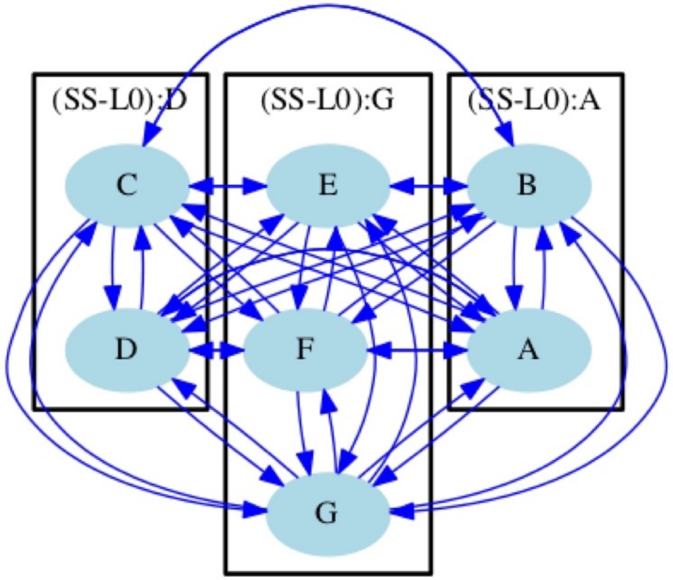


Microservices Identification from multiple processes



	A	B	C	D	E	F	G
A	0	2	0	1	0	0	0
B	0	0	2	0	0	0	0
C	0	0	0	1	0	0	0
D	0	0	0	0	1	1	0
E	0	0	0	1	0	0	1
F	0	0	0	0	0	0	0
G	0	0	1	0	0	0	0

	A	B	C	D	E	F	G
A	0	2	0.5	0.5	1	0	0
B	2	0	0.5	0.5	1	0	0
C	0.5	0.5	0	2	0.5	0.5	0
D	0.5	0.5	2	0	1	0.5	0
E	1	1	0.5	1	0	1.25	1
F	0	0	0.5	0.5	1.25	0	1
G	0	0	0	0	1	1	0

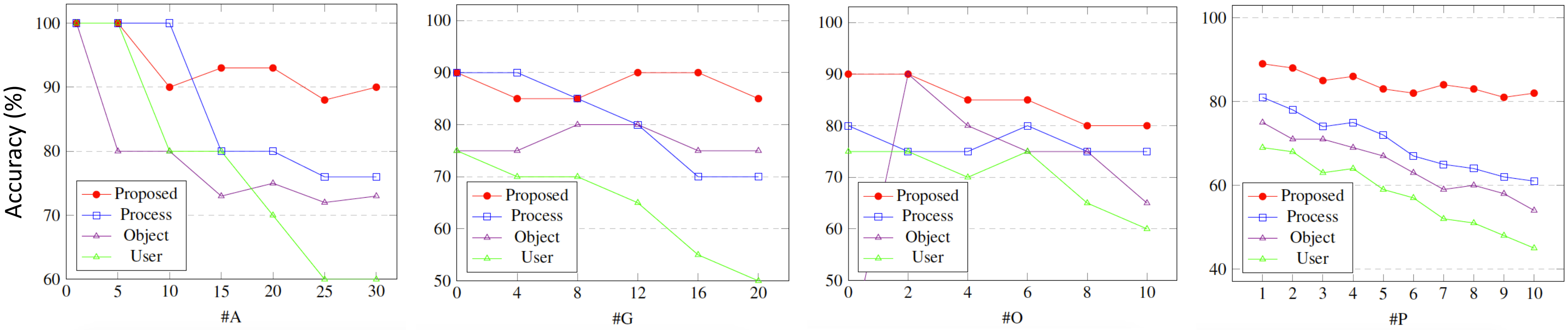


	Reads	Writes
A	O_1	O_1,O_2
B	O_1,O_2	O_1,O_2
C	O_2,O_3	O_3,O_4
D	O_1,O_3	O_3,O_4
E	O_3	O_1,O_5
F	O_3,O_5	O_5
G	O_5	O_5

Experiments

- Parameters: number of activities (#A), gateways (#G), objects (#O), and processes (#P).
- We measure Accuracy: the number of operations (activities) that are clustered in the correct microservice to the total number of operations.
- Correct microservices: we asked a group of domain experts to identify microservices
- Approaches:
  1) User-driven method: 3 developers are asked to identify microservices
  2) Process-driven: only the structural dependency of activities (relation T_p)
  3) Object-driven: only the data object dependency of activities (relation T_d)
  4) The proposed method: the combination of methods (2) and (3)

Experiments



Different number of activities, 1 ≤ #A ≤ 30 , but same number of gateways (#G = 4) and objects (#O = 3).

Different number of gateways, 0 ≤ #G ≤ 20 , but same number of activities (#A = 20) and objects (#O = 3).

Different number of objects, 0 ≤ #O ≤ 10 , but same number of activities (#A = 20) and gateways (#G = 4).

Different number of processes 1 ≤ #P ≤ 10 where on average for each process #A = 20, #G = 4, and #O = 3.

Conclusions

- Provide a method to identify an independent collection of highly inter-related activities as microservices.
- Measure dependencies between activities regarding their structural and data object interconnections
- The experiments show that the proposed method can identify cohesive, loosely coupled, and fine-grained microservices from a single business process, or a set of processes.
- The method can be easily generalized to other aspects such as requirements, resources, or ownerships.