2nd International Workshop on Decision Mining & Modeling for Business Processes (DeMiMoP'14)
in conjunction with BPM 2014 Eindhoven, The Netherlands, September 7th - 11th 2014

Constructing Probabilistic Process Models based on Hidden Markov Models for Resource Allocation

Berny Carrera, Jae-Yoon Jung

Department of Industrial and Management Systems Engineering

Kyung Hee University, Republic of Korea

Presented by Berny Carrera

Agenda

1. Introduction

1.1. Background

- Event logs are used in most organizations to store a large amount of information and allowing businesses to discover, analyze, model and make decisions based on the relations between events.
- Types of process mining
  - Discovery
  - Conformance
  - Enhancement
- The proposed approach considers the following perspectives:
  - The control flow perspective: Ordering of activities to explain the relations between the events.
  - The organizational perspective: Resource attributes to analyze how are they related with the events.

1.1. Background

Hidden Markov Model (HMM)

- HMM is a stochastic model that give us a direct probabilistic approach and can deal precisely with noise, incompleteness, and not over-fit the model.
- HMM can help describe how the resources are acting for resource allocation models, analyze the resource attribute of each event and the activities flow, and estimate the parameters for each activity (state) and resource (observation).

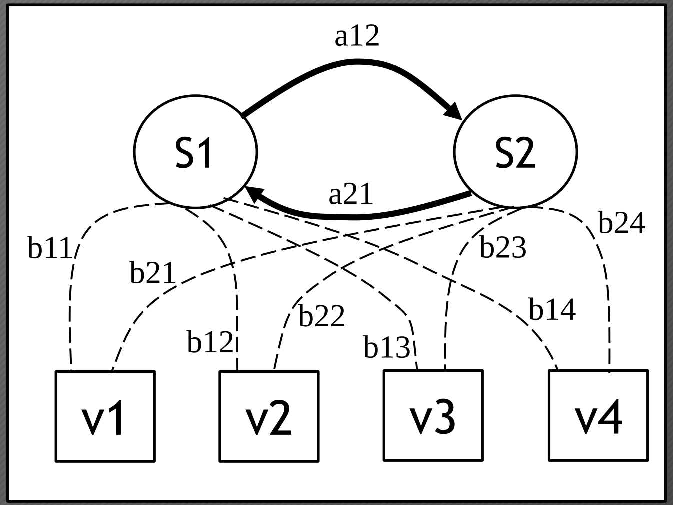


Fig. 1. Hidden Markov Model (Adapted from Baum, 1972)

1.2. Purpose

- This research evaluates the possibility of discovering and extending a probabilistic process model from a control-flow with an organizational perspective by applying HMM for resource allocation.
- Applicability of HMM for process mining by using event logs.
- Our approach allows us to answer related questions such as:
  - How to promote a resource effectively based on their experience?
  - How to determine the experience of resources in a specific activity?
  - What activity has to be scheduled and the resources required?

2. HMM-based Process Mining for Resource Allocation

7

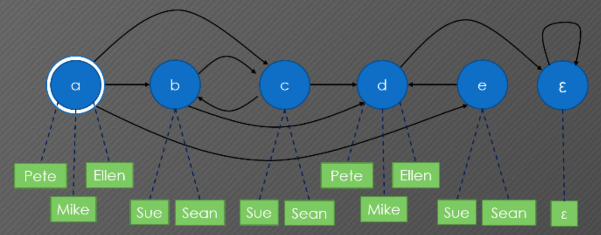
Overview of Resource Allocation Based on Hidden Markov Models

Event ID	Case ID	Activity	Resource
1	1	a	Pete
2	1	b	Sue
3	1	c	Sean
4	1	d	Mike
5	2	a	Ellen
6	2	b	Sean
7	2	c	Sue
8	2	d	Pete
9	3	a	Mike
10	3	b	Sue
11	3	c	Sue
12	3	d	Mike
13	4	a	Pete
14	4	c	Sean
15	4	b	Sue
16	4	d	Mike
17	5	a	Pete
18	5	c	Sean
19	5	b	Sue
20	5	d	Ellen
21	6	a	Ellen
22	6	e	Sue
23	6	d	Pete

Log

HMM-Miner Phase 1

- Construction of HMM-Workflow
  - Analysis of the Resource-Oriented Event Log
  - Discover a process model



HMM-Workflow

	a	b	c	d	e
a	#	→ , 3	→ , 2	#	→ , 1
b	←	#	|| , 3	→ , 2	#
c	←	|| , 2	#	→ , 3	#
d	#	←	←	#	←
e	←	#	#	→ , 1	#

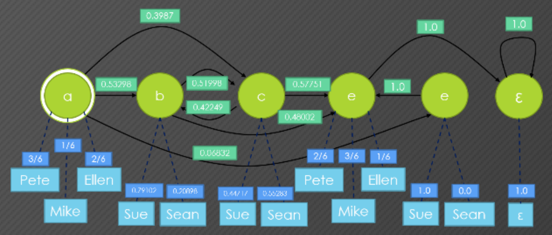
Footprint Matrix for HMM Miner

	Pete	Ellen	Sue	Mike	Sean
a	3	2	0	1	0
b	0	0	4	0	1
c	0	0	2	0	3
d	2	1	0	3	0
e	0	0	1	0	0

Frequency Matrix Fv

HMM-Miner Phase 2

- Calculation of the initial parameters
- Maximum Likelihood Estimation

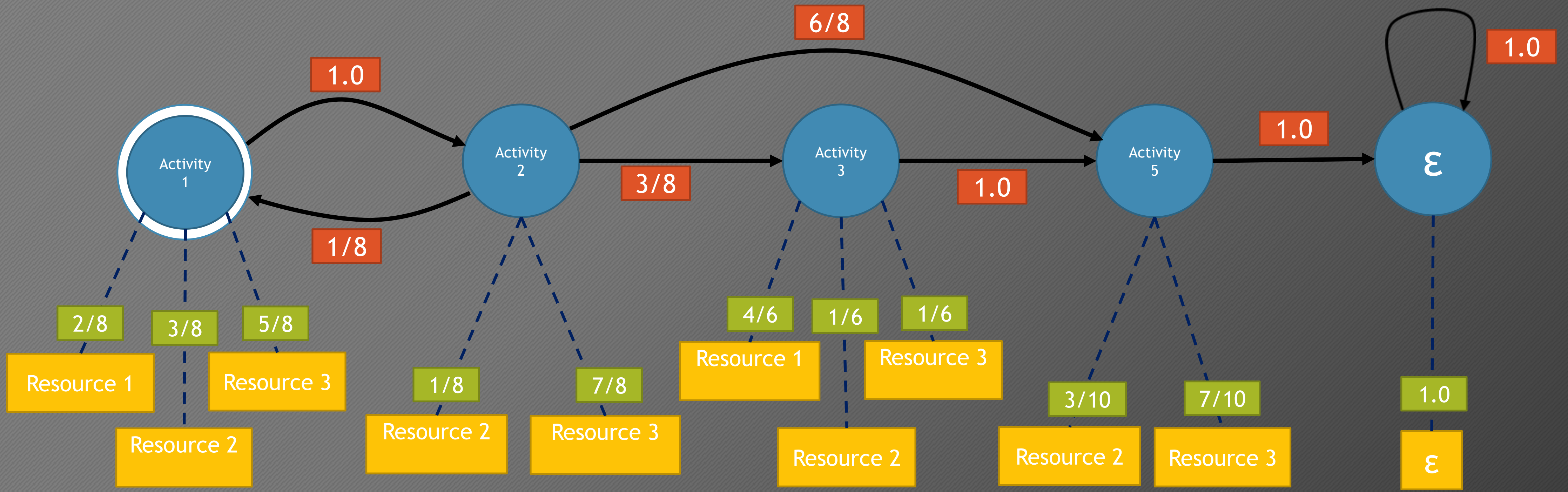


Estimated HMM-Workflow

2.1 HMM Miner – Phase 1

2.1.1. Construction of the HMM-Workflow

- To apply HMM for process mining, activities are represented by circles and resources are represented by squares.



2.1.1. Construction of the HMM-Workflow

- Formal Definitions

Definition 1. (Resource-Oriented Event Log) Let *L* be an event log. *L* is a tuple <*C*, *S*, *V*>, where *C* is the set of all possible cases, *S* of length *N* is the finite set of event labels $\{s_1, \ldots, s_N\}$ that has been performed over *L*, and $V \in L^*$ of length *M* is a set of originators $\{v_1, \ldots, v_M\}$ that specify the resource associated with the task.

Definition 2. (HMM Workflow) Let *L* be a resource-oriented event log. A HMM workflow $\theta(L) = (s, V, Fs, Fv)$ is represented by:

- $s = \{s_1, \ldots, s_N, \varepsilon_{N+1}\} \in L$ where $\{s_1, \ldots, s_N\}$ are the states or activities of *L* and $\{\varepsilon_{N+1}\}$ a dummy end state.
- $V = \{v_1, \ldots, v_M, \varepsilon_{M+1}\} \in L$ where $\{v_1 \ldots v_M\}$ are the observations of resources and $\{\varepsilon_{M+1}\}$ a dummy observation of the dummy end state $\{\varepsilon_{N+1}\}$.
- *Fs* is an *N*x*N* matrix with footprint sequence and frequency of occurrence for transitions s_i to s_j.
- *Fv* is an *N*x*M* matrix with the frequency of occurrence for resources *V* in state s_i.

Definition 3. (Footprint Matrix for HMM Miner) Let *L* be an event log over *£*, i.e. $L \in B(£^*)$. Let $x_1, x_2 \in £$:

- $x_1 >_L x_2$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \ldots, t_n \rangle$ and $i \in \{1, \ldots, n-1\}$ such that $\sigma \in L$ and $t_i = x_1$ and $t_{i+1} = x_2$ (Contains all pairs of activities in a "directly follows"). If is a directly follow, $fs = \sum_{\sigma \in L} L(\sigma)$
- $x_1 \rightarrow_L x_2$ if and only if $x_1 >_L x_2$ and $x_2 \not>_L x_1$ (Contains all pairs of activities in a "causality" relation
- $x_1 \#_L x_2$ if and only if $x_1 \not>_L x_2$ and $x_2 \not>_L x_1$
- $x_1 \|_L x_2$ of and only if $x_1 >_L x_2$ and $x_2 >_L x_1$

Resource-Oriented Event Log

HMM Workflow

Footprint Matrix for HMM Miner

2.1. HMM-Miner Phase 1

Resource-Oriented Event Log

- L_1 is a simple log describing the history of six cases.
- Our goal is discover a HMM that can "replay" event log L_1.

Event ID	Case ID	Activity	Resource
1	1	a	Pete
2	1	b	Sue
3	1	c	Sean
4	1	d	Mike
5	2	a	Ellen
6	2	b	Sean
7	2	c	Sue
8	2	d	Pete
9	3	a	Mike
10	3	b	Sue
11	3	c	Sue
12	3	d	Mike
13	4	a	Pete
14	4	c	Sean
15	4	b	Sue
16	4	d	Mike
17	5	a	Pete
18	5	c	Sean
19	5	b	Sue
20	5	d	Ellen
21	6	a	Ellen
22	6	e	Sue
23	6	d	Pete

2.1. HMM-Miner Phase 1

Event Log L_1

- $< a, b, c, d >^3$
- $< a, c, b, d >^2$
- $< a, e, d >$

- Find all activities *S* and all resources *V* in the traces from *L*.

$s = \{a, b, c, d, e\}$

V = {Pete, Ellen, Sue, Mike, Sean}

2.1. HMM-Miner Phase 1

Event Log L_1

- $< a, b, c, d >^3$
- $< a, c, b, d >^2$
- $< a, e, d >$

- Find start activities s_S and end activities s_e from L.

$s_S = \{a\}$

$s_e = \{d\}$

2.1. HMM-Miner Phase 1

Input

Event Log L_1

- $< a, b, c, d >^3$
- $< a, c, b, d >^2$
- $< a, e, d >$

- Create a footprint matrix for HMM Miner with frequencies *Fs* from *L*

	a	b	c	d	e
a	#	→	→	#	→
b	←	#	||	→	#
c	←	||	#	→	#
d	#	←	←	#	←
e	←	#	#	→	#

Footprint of L_1

	a	b	c	d	e
a	0	3	2	0	1
b	0	0	3	2	0
c	0	2	0	3	0
d	0	0	0	0	0
e	0	0	0	1	0

Frequency of the “directly follows” of L_1

Footprint Matrix for HMM Miner for L_1

Footprint of L_1

	a	b	c	d	e
a	#	→	→	#	→
b	←	#	||	→	#
c	←	||	#	→	#
d	#	←	←	#	←
e	←	#	#	→	#

Frequency of the "directly follows" of L_1

	a	b	c	d	e
a	0	3	2	0	1
b	0	0	3	2	0
c	0	2	0	3	0
d	0	0	0	0	0
e	0	0	0	1	0

Output

	a	b	c	d	e
a	#	→ , 3	→ , 2	#	→ , 1
b	←	#	|| , 3	→ , 2	#
c	←	|| , 2	#	→ , 3	#
d	#	←	←	#	←
e	←	#	#	→ , 1	#

Footprint Matrix for HMM Miner for L_1

2.1. HMM-Miner Phase 1

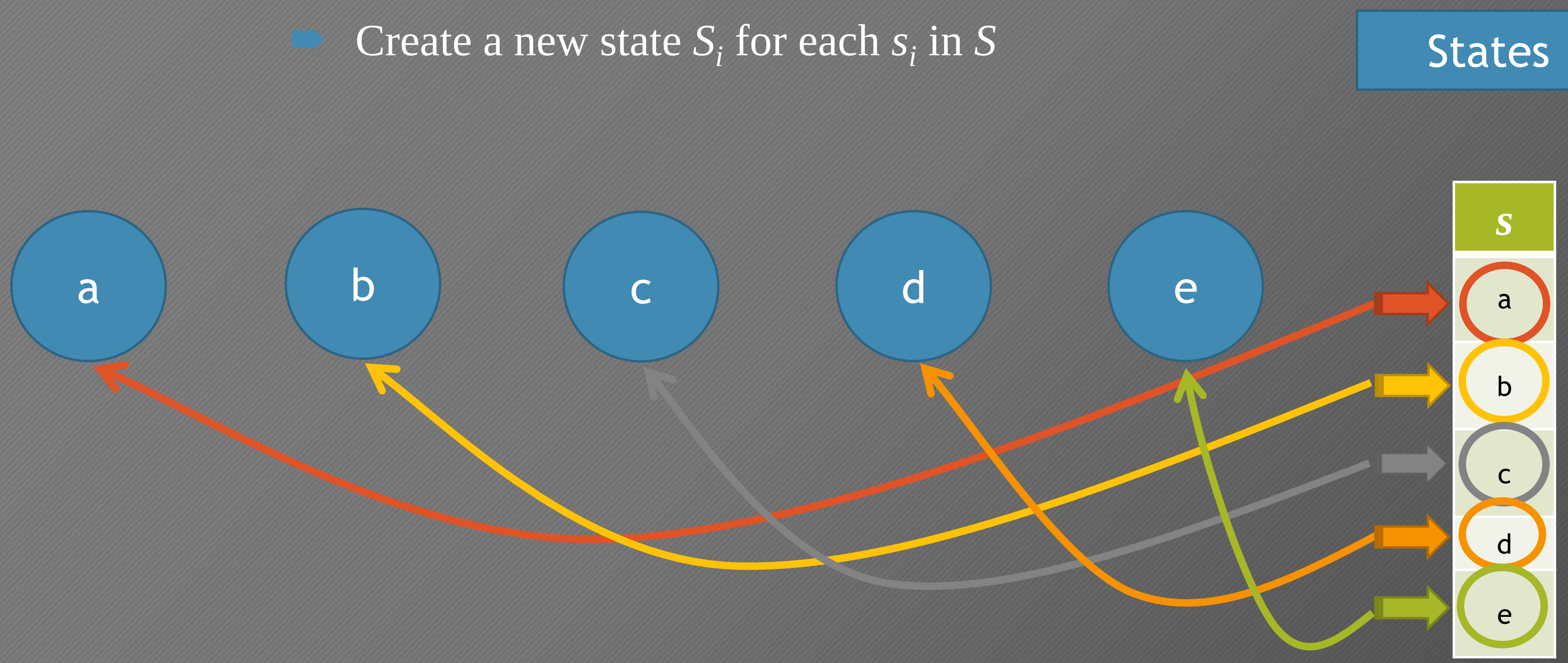
- Create a Frequency Matrix F_v for each s_i in S with the resources in V.

Frequency Matrix F_v with the occurrence for resources V by activity.

	Pete	Ellen	Sue	Mike	Sean
a	3	2	0	1	0
b	0	0	4	0	1
c	0	0	2	0	3
d	2	1	0	3	0
e	0	0	1	0	0

2.1. HMM-Miner Phase 1

- Create a new state S_i for each s_i in S



2.1. HMM-Miner Phase 1

- Add the corresponding state s_i into the initial state set S_S with its starting probability π.

s_S
a

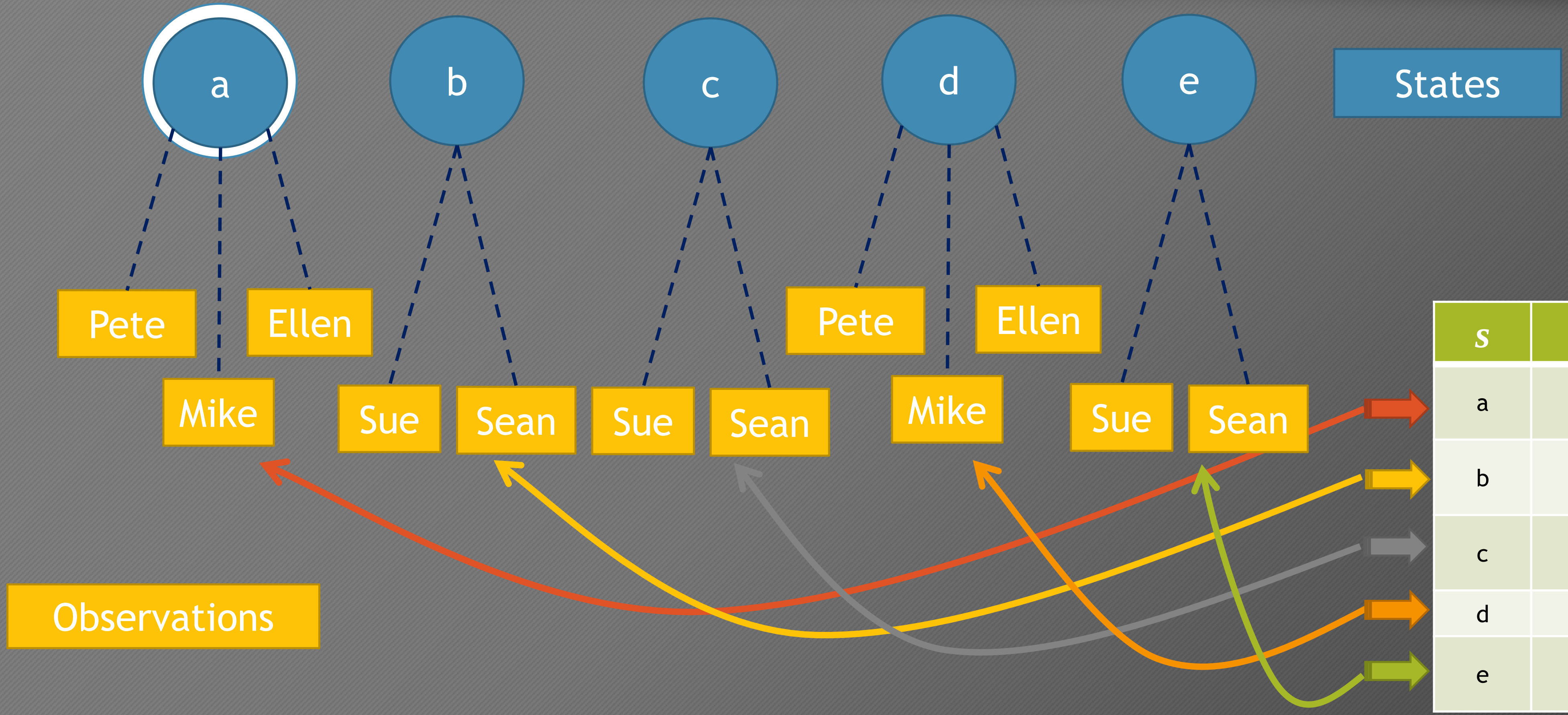
Initial state set

Initial state	Probability
a	1

$\pi_i = [1,0,0,0,0,0]^{T}$

2.1. HMM-Miner Phase 1

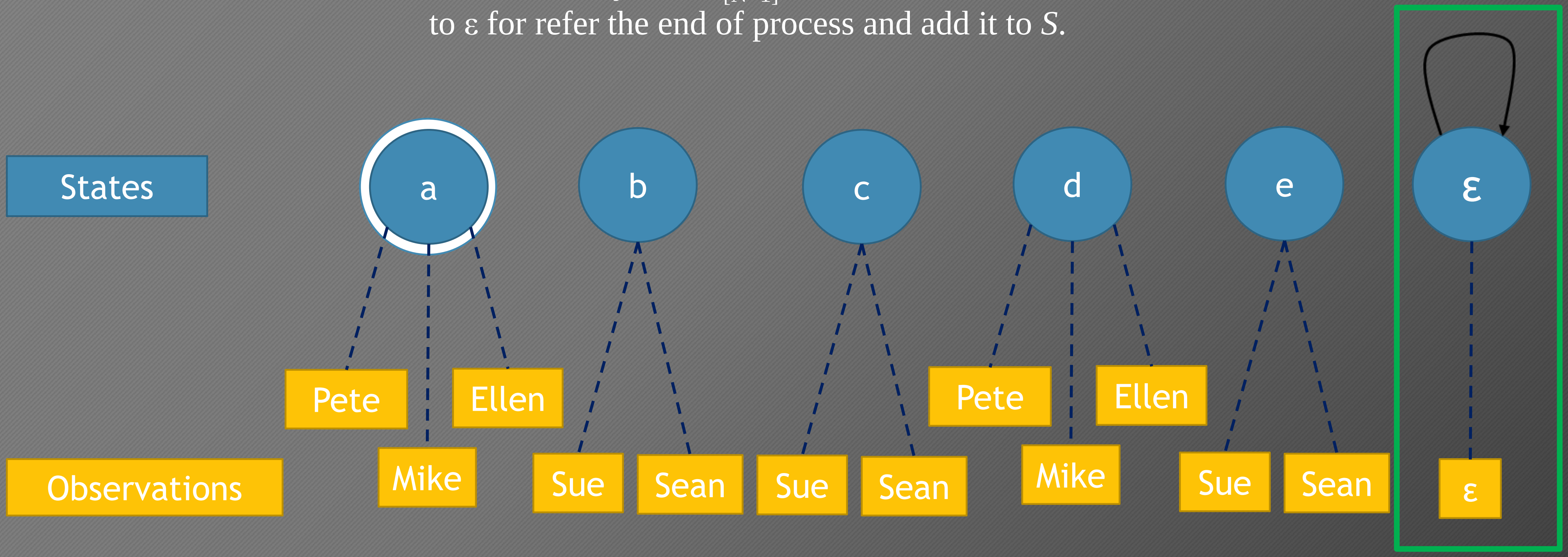
- Create an arc from s_i to v_j



s	*Resource*
a	Pete, Mike, Ellen
b	Sue, Sean
c	Sue, Sean
d	Pete, Mike, Ellen
e	Sue, Sean

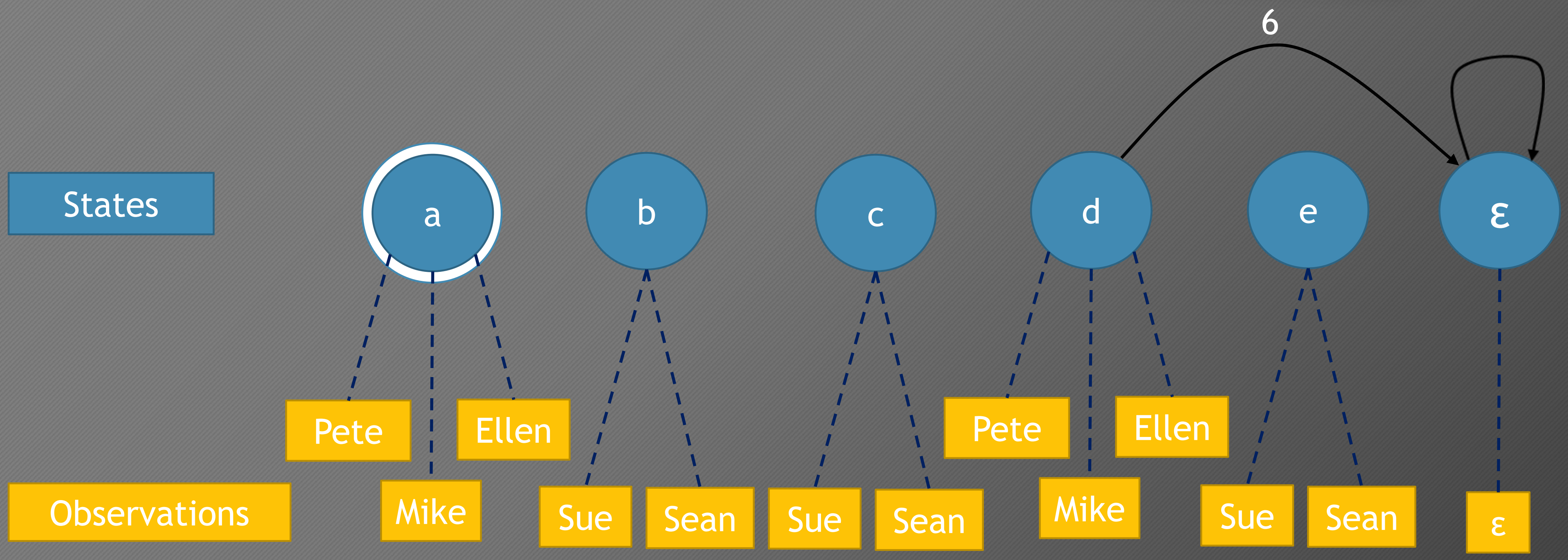
2.1. HMM-Miner Phase 1

- Create dummy state $s_{[N+1]} = \varepsilon$ with an arc from ε to ε for refer the end of process and add it to *S*.



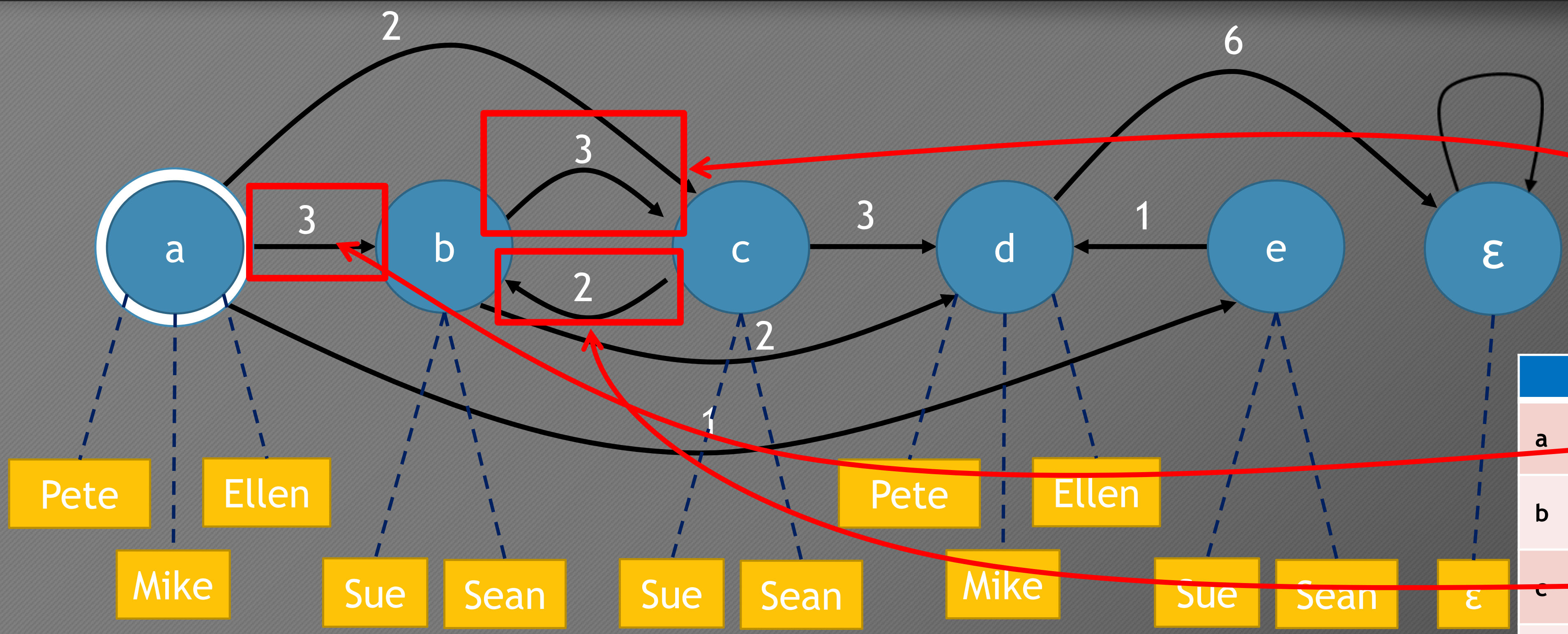
2.1. HMM-Miner Phase 1

- For all end activities given by s_E the corresponding states s_i are end states; then create an arc from s_i to ε and calculate each frequency



2.1. HMM-Miner Phase 1

- If fs_{ij} in Fs has a “direct follow” then
  - create an arc from s_i to s_j with its frequency

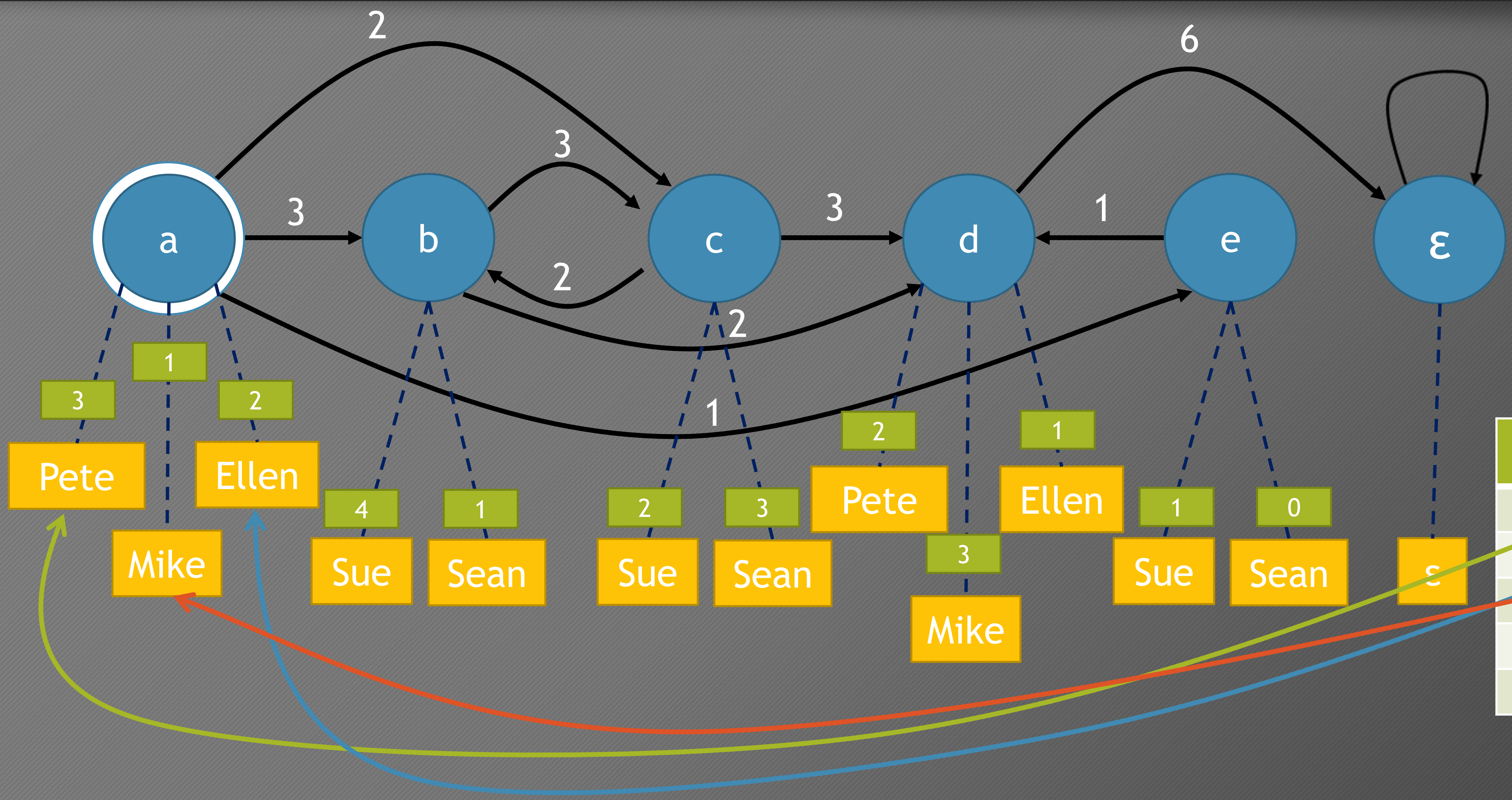


	a	b	c	d	e
a	#	→ , 3	→ , 2	#	→ , 1
b	←	#	|| , 3	→ , 2	#
c	←	|| , 2	#	→ , 3	#
d	#	←	←	#	←
e	←	#	#	→ , 1	#

Footprint for HMM Matrix for L1

2.1. HMM-Miner Phase 1

- If fv_{ij} in F has a "direct follow" then
  - create an arc from s_i to v_j with its frequency



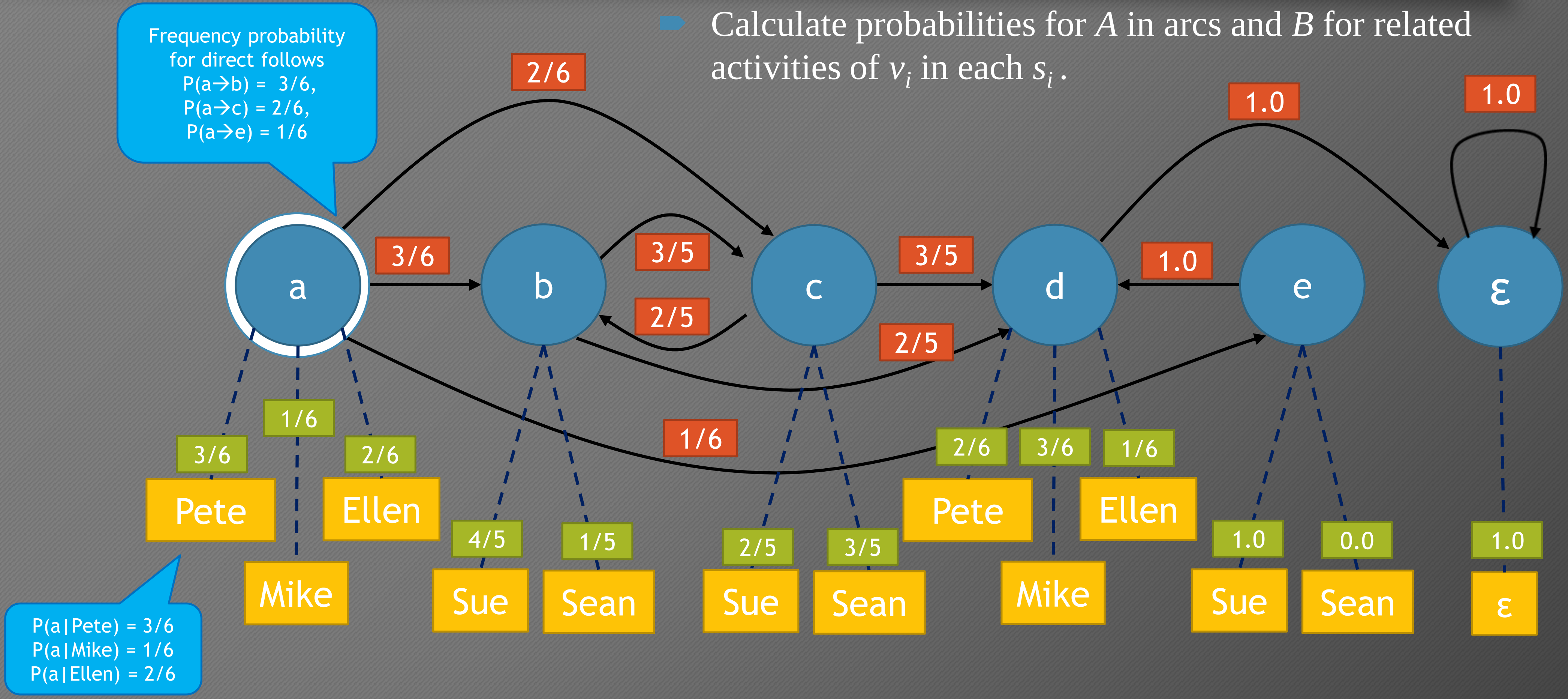
	Pete	Ellen	Sue	Mike	Sean
a	3	2	0	1	0
b	0	0	4	0	1
c	0	0	2	0	3
d	2	1	0	3	0
e	0	0	1	0	0

Frequency Matrix F_v for L1

2.2. HMM Miner – Phase 2

2.2.1. Initial parameters of the HMM

- Calculate probabilities for A in arcs and B for related activities of v_i in each s_i .



2.2.1. Initial parameters of the HMM

Initial parameters of the HMM for L_1

- Number of states (activities): N = 6
- Number of observations (resources) : 6
  - V = {Pete, Mike, Ellen, Sue , Sean, ε}
- Initial state distribution $\pi_i = P\ (\ q_1 = i\ ),\ = [1,0,0,0,0,0]^T$
- State transition probability distribution

$A = \{a_{ij}\} =$

	a	b	c	d	e	ε
a	0	3/6	2/6	0	1/6	0
b	0	0	3/5	2/5	0	0
c	0	2/5	0	3/5	0	0
d	0	0	0	0	0	1
e	0	0	0	1.0	0	0
ε	0	0	0	0	0	1.0

- Observation symbol probability distribution

$B = \{b_i(v_k)\} =$

	Pete	Mike	Ellen	Sue	Sean	ε
a	3/6	1/6	2/6	0	0	0
b	0	0	0	4/5	1/5	0
c	0	0	0	2/5	3/5	0
d	2/6	3/6	1/6	0	0	0
e	0	0	0	1.0	0	0
ε	0	0	0	0	0	1.0

2.2.2. Maximum Likelihood Estimation

- Expectation Maximization (EM) algorithm is used to estimate the maximum likelihood in selecting the best values for the model parameters that make the observed data the most probable.
- In real-world scenarios EM algorithm can deal precisely with noise and incompleteness.

```
Inputs: λ = ( N, M, A, B, π ) and event log L
Initialize
Repeat
    Using forward algorithm calculate α_i for each trace in L
    Using backward algorithm calculate β_i for each trace in L
    Re-estimation of temporal variables
            Calculate γ and ξ based on ∑ α_i and ∑ β_i
            Calculate A*, B*, π* from the temporal variables
            Updating λ
Until A and B do not change
Output: λ* = ( N, M, A*, B*, π* )
```

Fig. 3. Expectation-Maximization procedure.

Trained Parameters for the HMM for L_1

- Number of states (activities): N = 6
  - s={a, b, c, d, e, ε}
- Number of observations (resources) : 6
  - V = {Pete, Mike, Ellen, Sue , Sean, ε}
- Initial state distribution
  - $\pi_i = P\ (\ q_1 = i\ ),\ = [1,0,0,0,0,0]^T$
- State transition probability distribution

$A^* = \{a_{ij}\} =$

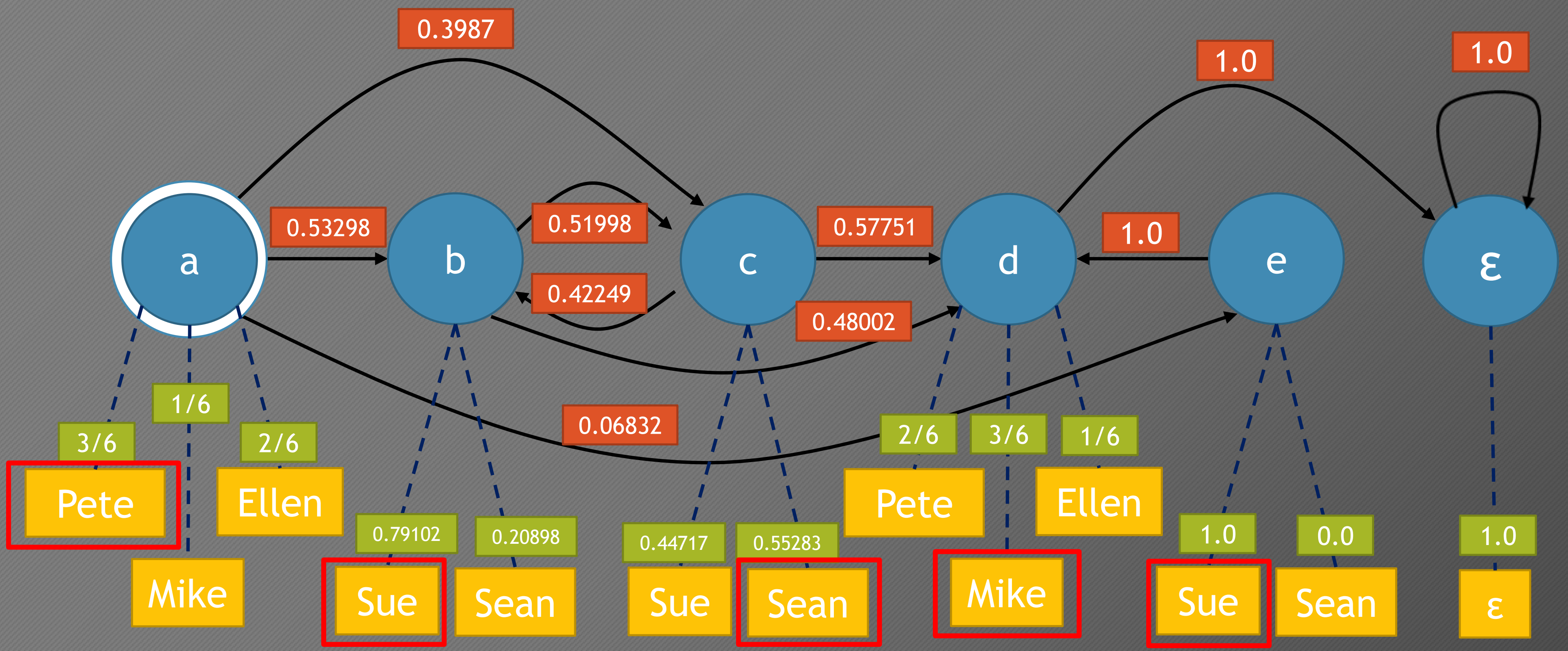
	a	b	c	d	e	ε
a	0	0.53298	0.3987	0	0.6832	0
b	0	0	0.51998	0.48002	0	0
c	0	0.42249	0	0.57751	0	0
d	0	0	0	0	0	1.0
e	0	0	0	1.0	0	0
ε	0	0	0	0	0	1.0

- Observation symbol probability distribution

$B^* = \{b_i(v_k)\} =$

	Pete	Mike	Ellen	Sue	Sean	ε
a	0.5	0.16667	0.33333	0	0	0
b	0	0	0	0.79102	0.20898	0
c	0	0	0	0.44717	0.55283	0
d	0.33333	0.5	0.16667	0	0	0
e	0	0	0	1.0	0	0
ε	0	0	0	0	0	1.0

Proposed Model for HMM Miner from L_1



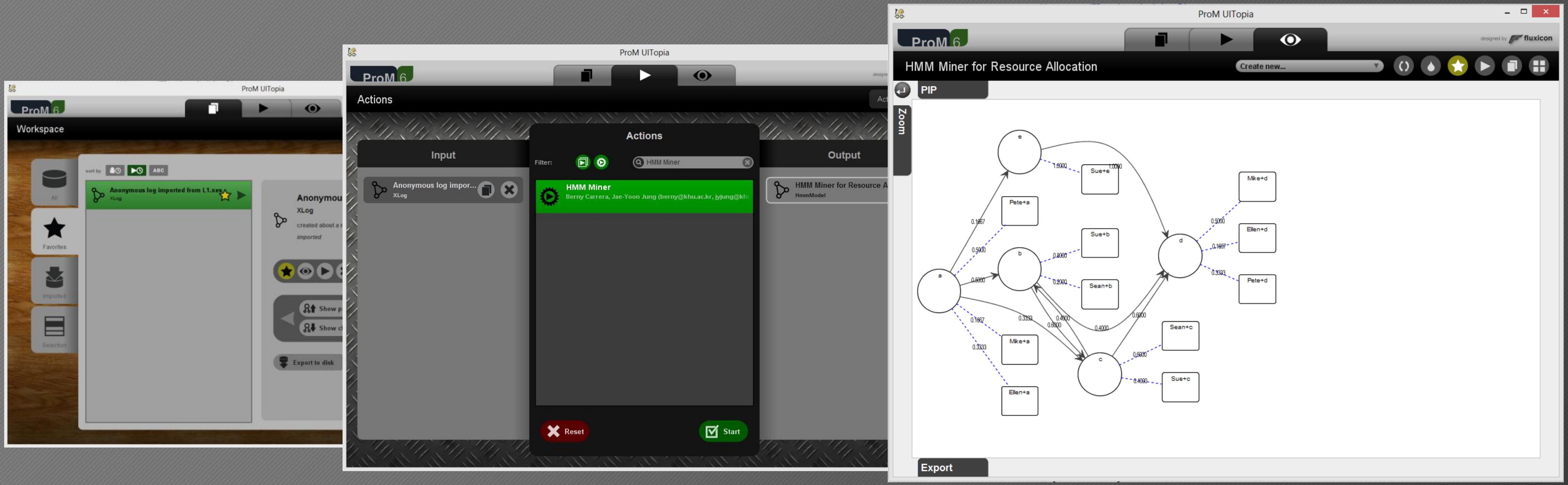
Event Log L1

- $< a, b, c, d >^3$
- $< a, c, b, d >^2$
- $< a, e, d >$

3. Implementation

3. HMM Miner - Implementation

- The technique presented in this paper was implemented as a plug-in for the ProM Framework and is called HMM Miner.



4. Conclusion

4. Conclusion

- Summary
  - HMM-Miner designs a probabilistic discovery process from event logs using HMM to support resource allocation.
  - Expectation Maximization approach was adopted to estimate the model parameters and is useful in real-world scenarios to manage standard errors and noise.
  - Since determining the number of hidden states is very difficult, the model is based on activities and resources in such a way that the comprehension of the model is enhanced.
- Contribution
  - A process discovery method that combines an organizational perspective with a probabilistic approach to address the resource allocation and improve the productivity of resource management.
  - The proposed approach is helpful to compare the performance of resources for activity executions.

4. Conclusion

- Limitation and Future Work
  - Application with Viterbi Algorithm for finding the most likely sequence in the activities for online process mining.
  - Consideration of the time perspective to analyze if a resource is busy and the extension of the technique to other scenarios.
  - Evaluate the proposed approach using real-life event logs.