POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale – Classe LM-31 (DM 270)
Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

Limiti del Process Mining in ambito produttivo

Relatore
Prof. Arianna Alfieri

Candidato
Davide Diana

Anno accademico 2018-2019

Sommario

Introduzione

I sistemi informativi aziendali sono sempre più intrecciati con i processi operativi a cui forniscono supporto. Negli ultimi decenni ci sono stati importanti progressi tecnologici. La vita delle persone e delle organizzazioni ha vissuto un cambiamento epocale con l'avvento di internet; basti pensare che in tempi moderni persone e aziende ricorrono con grande frequenza a dispositivi digitali e a informazioni provenienti dalla rete. La logica conseguenza di questo scenario affermatosi recentemente consiste nella presenza di un'abbondante quantità di dati, dei quali non era possibile disporre in periodo predigitale. In ambito d'impresa, molti dati si riferiscono ai processi operativi e per tale ragione si tiene traccia di essi all'interno dei sistemi informativi; tuttavia, sebbene il loro reperimento non sia complicato, le organizzazioni trovano difficoltà ad interpretare i dati e a trarne informazioni utili al loro business. In tempi molto recenti e con contorni non ancora perfettamente delineati, si è affacciata sulla scena industriale e gestionale una nuova disciplina, nota come Process Mining; si tratta di un nuovo strumento del quale le aziende potranno avvalersi per sanare le loro carenze riguardo l'interpretazione dei dati e per accrescere le loro competenze. Il contributo che il process mining sarà in grado di fornire si spera possa essere efficace in misura simile all'effetto rivoluzionario che ebbe internet in seguito alla sua comparsa nel mondo.

Secondo quanto afferma Wil van der Aalst, professore del Department Mathematics & Computer Science presso la University of Technology di Eindhoven, "Process mining, i.e., extracting valuable, process-related information from event logs, complements existing approaches to Business Process Management (BPM)"[1]. Il BPM è la disciplina che combina le conoscenze gestionali con l'Information Technology (IT) e si applica nell'ambito di processi operativi aziendali. Si tratta di una materia che in epoca recente ha generato molta attenzione: potenzialmente, è in grado di garantire un aumento della produttività e un significativo risparmio sui costi. Il BPM può essere visto come un'estensione del Workflow Management (WFM), il quale si concentra prevalentemente sull'automazione dei processi. Il BPM, invece, opera su un campo più ampio, spaziando dall'automazione e dall'analisi dei processi alla gestione dei processi stessi e all'organizzazione del lavoro. Il BPM si pone come obiettivo il miglioramento dei processi operativi aziendali, possibilmente senza l'uso di nuove tecnologie; tuttavia, spesso il BPM è accompagnato da un software avente il compito di gestire, controllare e supportare i processi operativi. Il professor van der Aalst include i tradizionali sistemi WFM all'interno dei Process-Aware Information Systems (PAIS), i quali comprendono anche sistemi in grado di fornire più flessibilità ai processi o di espletare compiti specifici; ad esempio, si annoverano fra questi i più grandi sistemi ERP (Enterprise Resource Planning) quali Oracle e SAP, i sistemi CRM (Customer Relationship Management),

[1] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, p. 3. ["Il process mining, ossia l'estrazione di informazioni relative ai processi dai log di eventi, completa gli attuali approcci al Business Process Management (BPM)"].

i sistemi basati su regole ben definite (rule-based), ecc. Il BPM e i sistemi PAIS si fondano entrambi sui modelli di processo (process models). Sono parecchie le notazioni che permettono alle aziende di modellare i loro processi operativi (ad esempio reti di Petri e BPMN) e tutte hanno come caratteristica comune la descrizione dei processi in termini di attività, il cui ordinamento è determinato dalle dipendenze esistenti fra di esse.

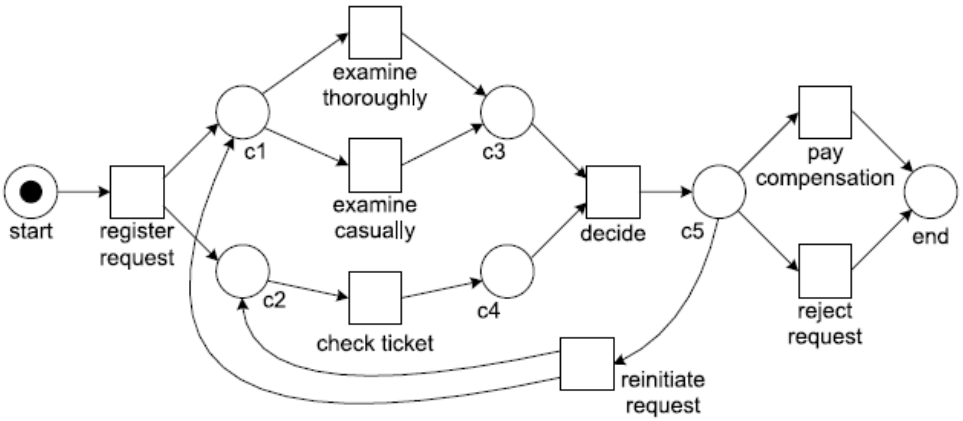

Figura 1. Un esempio di rete di Petri per la gestione di una richiesta di risarcimento.

In Figura 1. è mostrato un esempio di modello di processo rappresentato con una rete di Petri, la quale è definita con una tripla *(P, T,F)*. *P* è un insieme finito di posti, *T* è un insieme finito di transizioni *(P ∩ T = ∅)*, mentre *F ⊆ (P × T) ∪ (T × P)* è un insieme di archi orientati i quali collegano posti e transizioni. Ogni transizione è rappresentata da un quadrato e ogni posto è rappresentato da un cerchio. I posti non sono altro che gli stati assumibili dal processo. Una transizione è abilitata, cioè può aver luogo la corrispondente attività, solo se tutti i posti in ingresso contengono un token; la condizione appena menzionata, necessaria per l'abilitazione della transizione, prende il nome di regola di scatto (oppure firing in inglese). Una transizione che soddisfa la condizione di firing consuma i token presenti nei suoi posti in ingresso e ne produce uno per ciascuno dei suoi posti in uscita. La configurazione assunta dai token lungo i molteplici posti della rete prende il nome di marcatura (o marking in inglese). I token sono rappresentati da pallini neri[2].

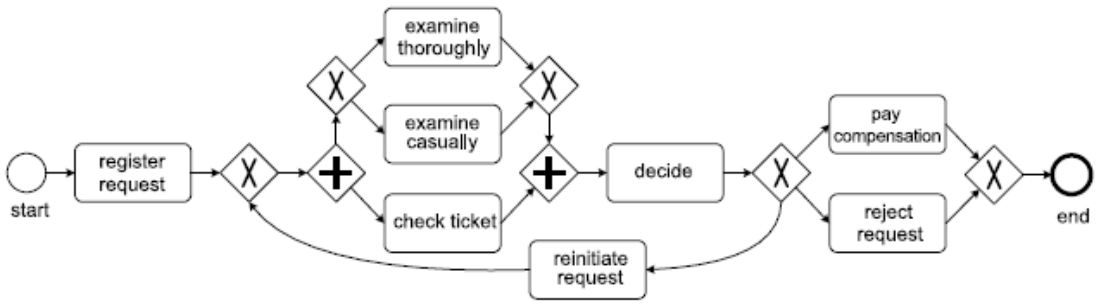

Figura 2. Lo stesso processo modellato con la notazione BPMN.

[2] W.M.P. van der Aalst, op. cit., p. 4.

Nella Figura 2. è invece rappresentato un esempio di modello di processo descritto con la notazione BPMN (Business Process Modeling Notation), la quale sostituisce i posti con delle porte di accesso, dette gateway. I rombi con all'interno il segno x sono gateway di tipo OR, mentre quelli con all'interno il segno + sono gateway di tipo AND; entrambe le tipologie di gateway possono essere congiungenti o disgiungenti, a seconda del numero di archi in ingresso e in uscita. A parte questo, il comportamento del diagramma BPMN è il medesimo della rete di Petri descritta in precedenza[3].

Le Figure 1. e 2. mostrano il modello solo nell'ottica del flusso di controllo, ossia della sequenza con cui sono eseguite le attività di processo; tuttavia, i linguaggi di modellazione non si limitano al flusso di controllo ma consentono di descrivere il processo tramite modelli che considerano altre prospettive, alcune delle quali sono illustrate in seguito nell'elaborato.

In questo contesto, come evidenzia van der Aalst, sono diversi i motivi per i quali si utilizzano i modelli di processo[4]; i principali sono elencati di seguito.

- Comprensione: chi realizza il modello ha la possibilità di vedere il processo da varie angolazioni.
- Discussione: gli stakeholder dell'organizzazione usano i modelli per preparare le loro discussioni.
- Documentazione: i processi sono documentati per istruire le persone e per fini di certificazione (es. ISO 9000 sulla gestione per la qualità).
- Verifica: i modelli di processo sono analizzati per individuare errori all'interno dei sistemi o delle procedure.
- Analisi di prestazione: le tecniche come la simulazione possono essere usate per cogliere i fattori che influenzano le misure di performance quali i tempi di risposta, i livelli di servizio, ecc.
- Animazione: i modelli permettono agli utenti di sperimentare scenari differenti traendo così dei feedback a vantaggio del progettista.
- Specificazione: l'uso dei modelli è utile a descrivere un sistema PAIS prima che sia implementato e, pertanto, essi sono anche una sorta di contratto fra il programmatore, il quale deve attenersi alle specifiche, e l'utilizzatore finale.
- Configurazione: i modelli possono essere utilizzati per configurare un sistema.

Come è possibile immaginare, i modelli di processo giocano un ruolo significativo specie nelle grandi aziende. Il professor van der Aalst spiega che, solitamente, sono due le tipologie di modello alle quali fare ricorso: modelli informali e modelli formali. I modelli informali si adottano per la discussione e la documentazione mentre i modelli formali sono adoperati per l'analisi o l'esecuzione dei processi. Per caratteristica, i modelli informali sono vaghi e ambigui; da contraltare, i modelli formali tendono a

[3] W.M.P. van der Aalst, op. cit., p. 5.
[4] W.M.P. van der Aalst, op. cit., p. 6.

concentrarsi su confini molto più ristretti o ad essere dettagliati a tal punto da risultare incomprensibili per gli stakeholder. Wil van der Aalst sostiene:“ Independent of the kind of model—informal or formal—one can reflect on the alignment between model and reality”[5]. Un modello di processo usato per configurare un sistema di workflow management probabilmente è ben allineato con la realtà, dato che, per via del modello, le persone sono quasi obbligate a lavorare in un determinato modo. Purtroppo, la maggior parte dei modelli realizzati a mano sono distaccati dalla realtà e offrono solamente una visione astratta e idealizzata dei processi in questione. Inoltre, anche i modelli formali, che permettono di condurre analisi rigorose, possono trovare difficoltà a riflettere fedelmente i processi attuali. Per cui, affinché i modelli siano strumenti validi e dall'indubbio valore tecnico, strategico e gestionale, è necessario prestare grande attenzione all'allineamento con la realtà. Non avrebbe senso ad esempio sperimentare un modello che assume una versione non verosimile o addirittura irreale di un processo. Un sistema implementato sulla base di modelli idealizzati è probabile che crei disturbo e non sia tollerato dai clienti finali. In aggiunta, i principali modelli di riferimento presentano una qualità non adeguata ad assolvere il proprio compito in maniera corretta. Questo aspetto fa sì che non ci sia un buon allineamento fra modello e realtà. Se però si mette in relazione il problema dell'allineamento con l'abbondanza di dati relativi agli eventi, è possibile ampliare la conoscenza sui processi attuali ma anche valutare e migliorare i modelli di processo esistenti, entrambi obiettivi che il process mining si prefigge.

Il process mining è una disciplina organizzativa, particolarmente innovativa, i cui studi di ricerca iniziarono nel 1999 presso la University of Technology di Eindhoven, in Olanda, alla quale si è aggiunta pochi anni dopo la Queensland University of Technology di Brisbane, in Australia. A quei tempi, non vi era grande disponibilità di dati e le tecniche di process mining che si riuscirono a sviluppare erano alquanto primitive nonché inutilizzabili. Nei successivi 10-15 anni, avendo a disposizione molti più dati rispetto al passato, le tecniche sono maturate e migliorate. Il process mining costituisce pertanto un'area di ricerca relativamente giovane che si trova, da un lato, tra il machine learning ed il data mining e, dall'altro, tra la modellazione e l'analisi dei processi. L'idea di base del process mining, come spiega van der Aalst, consiste nel dedurre, monitorare e migliorare i processi reali (cioè attuali e non ipotetici) estraendo conoscenza dai log, i quali rappresentano gli eventi registrati all'interno dei sistemi informativi aziendali e che, come già accennato, oggi sono disponibili in grandi quantità[6].

[5] W.M.P. van der Aalst, op. cit., p. 6. [“Indipendentemente dalla tipologia, formale o informale, occorre riflettere sull'allineamento fra il modello e la realtà”].

[6] W.M.P. van der Aalst, op. cit., p. 8

Sempre van der Aalst, nella prefazione del suo libro, delinea il contesto che ha portato alla nascita del process mining:

> *There are two main drivers for this new technology. On the one hand, more and more events are being recorded thus providing detailed information about the history of processes. Despite the omnipresence of event data, most organizations diagnose problems based on fiction rather than facts. On the other hand, vendors of Business Process Management (BPM) and Business Intelligence (BI) software have been promising miracles. Although BPM and BI technologies received lots of attention, they did not live up to the expectations raised by academics, consultants, and software vendors*[7].

Per posizionare meglio il process mining occorre descrivere il ciclo di vita BPM (Figura 3.), costituito da diverse fasi per la gestione di un particolare processo di business. Nella fase design si progetta il processo. Il modello ottenuto passa poi alla fase configuration/implementation. Nel caso in cui il modello sia già in forma eseguibile, tale fase sarà molto rapida; tuttavia, se il modello dovesse richiedere la trascrizione in un linguaggio compatibile con un software convenzionale, potrebbe richiedere più tempo.

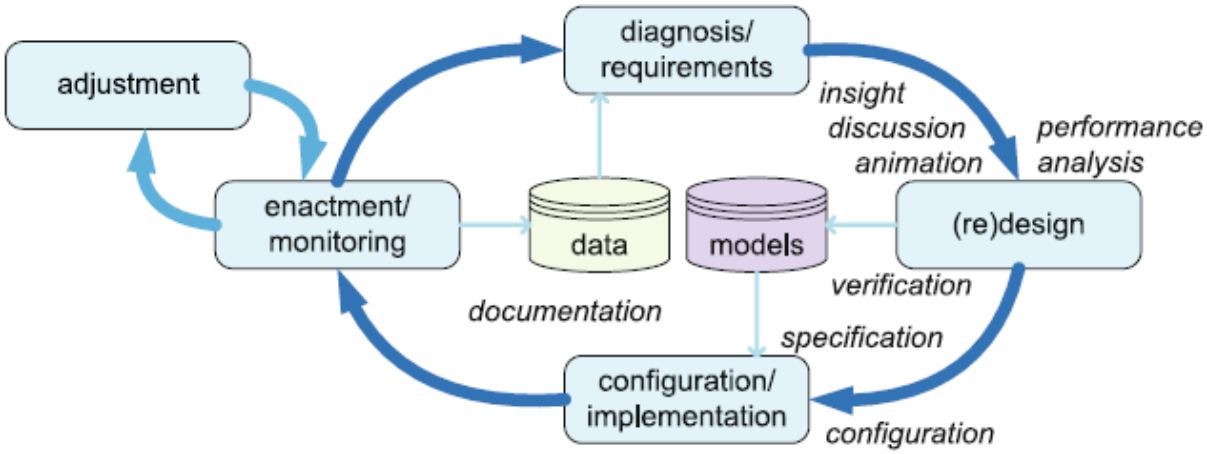

Figura 3. Il ciclo di vita BPM.

Terminata l'implementazione, il sistema è in grado di supportare i modelli di processo precedentemente disegnati. Pertanto, può iniziare la fase enactment/monitoring, nella quale il management si occupa di monitorare i processi in esecuzione per valutare se sia necessario attuare modifiche. Alcuni degli eventuali correttivi sono trattati e apportati nella fase adjustment, nella quale non si ridisegna il processo né si creano nuovi software ma si utilizzano solamente comandi predefiniti

[7] W.M.P. van der Aalst, op. cit., preface. ["Sono due le principali ragioni che hanno portato a sviluppare questa nuova tecnologia. Da un lato, sempre più eventi sono stati registrati al fine di acquisire informazioni dettagliate circa i comportamenti storici dei processi. Nonostante la presenza costante di innumerevoli dati, la maggior parte delle aziende si basa sulla teoria per rilevare problemi ai loro processi produttivi invece che sui fatti. Dall'altro, i produttori di software per il Business Process Management (BPM) e il Business Intelligence (BI) hanno promesso soluzioni molto efficaci e performanti. Sebbene queste tecnologie ricevettero grande attenzione, non sono riuscite a rispettare le attese sperate da consulenti, accademici e dai produttori stessi"].

per adattare o riconfigurare il processo. In seguito, vi è la fase diagnosis/requirements, utile a valutare il processo e a monitorare l'insorgere di nuovi requisiti dovuti al mutamento dell'ambiente in cui opera il processo stesso (es. competizione, nuove leggi); nel caso si riscontrassero bassi livelli di performance o nuove richieste imposte dall'ambiente esterno, il ciclo sarà sottoposto ad una nuova iterazione, a partire dalla fase redesign. I modelli di processo assumono un ruolo predominante nelle fasi design e configuration/implementation; parimenti i dati nelle fasi enactment/monitoring e diagnosis/requirements. In passato vi erano poche connessioni tra i dati prodotti durante l'esecuzione del processo e il modello di processo. Infatti, nella maggioranza delle imprese la fase diagnosis/requirements non era supportata in modo continuo e sistematico. In particolare, solamente in seguito a gravi problemi o a grandi cambiamenti esterni si procedeva a reiterare il ciclo di vita BPM; inoltre, le reali informazioni sull'attuale processo non erano prese in considerazione nell'ambito delle decisioni di redesign[8].

A valle di queste considerazioni, si può asserire che il process mining sia l'elemento che porta a pieno compimento il ciclo BPM. I dati registrati dai sistemi informativi sono funzionali a ricavare una visuale migliore sui processi attuali, in modo che le eventuali deviazioni possano essere analizzate e la qualità dei modelli possa essere migliorata. In quest'ottica, il process mining assolve il compito di collegare i processi attuali e i loro dati con i modelli di processo. Come già anticipato, si fa riferimento ai dati con il termine log di eventi; tali dati, conservati all'interno dei sistemi informativi, devono essere estratti, ossia si deducono informazioni riguardo il funzionamento effettivo dell'organizzazione. Qualsiasi sforzo profuso legato al process mining vede l'estrazione dei dati come una sua parte integrante. È importante che gli eventi registrati nei log siano univoci e discernibili fra loro; a tal proposito, solitamente si conservano all'interno dei log anche informazioni aggiuntive circa gli eventi. Infatti, ogni qualvolta sia possibile, il process mining si serve di informazioni extra come la risorsa (persona o dispositivo) che ha iniziato o eseguito una certa attività, le coordinate temporali in cui è avvenuta una registrazione, o altri elementi correlati con l'evento in esame (es. la dimensione di un ordine). Il professor van der Aalst rileva che i log di eventi, come si può osservare in Figura 4., possono essere sfruttati per condurre tre diverse tipologie di Process Mining[9]:

1. Process Discovery: una tecnica di discovery, partendo da un log di eventi, produce un modello senza utilizzare informazioni a priori. Un esempio è rappresentato dall'algoritmo α, il quale considera un log di eventi e crea una rete di Petri con la quale spiega i comportamenti registrati all'interno del log.
2. Conformance Checking: il modello di un processo esistente è comparato con il log di eventi dello stesso processo. Il conformance checking può essere usato per verificare se la realtà, così come

[8] W.M.P. van der Aalst, op. cit., pp. 7-8.
[9] W.M.P. van der Aalst, op. cit., p. 10.

registrata nel log, si conforma al modello e viceversa. Per esempio, si può considerare un modello di processo che per l'emissione di ordini di valore superiore a 1M€ richiede due controlli. L'analisi del log mostrerà se questa regola è correttamente rispettata o meno. Un altro esempio è il controllo del cosiddetto principio "four-eyes", secondo il quale le attività più particolari non dovrebbero essere eseguite né da una sola né dalla stessa persona. Esaminando il log tramite l'uso del modello, al cui interno sono specificati questi requisiti, è possibile accorgersi di eventuali inosservanze. Perciò, il conformance checking si presta bene a rilevare, localizzare e successivamente spiegare le deviazioni e misurarne la gravità.

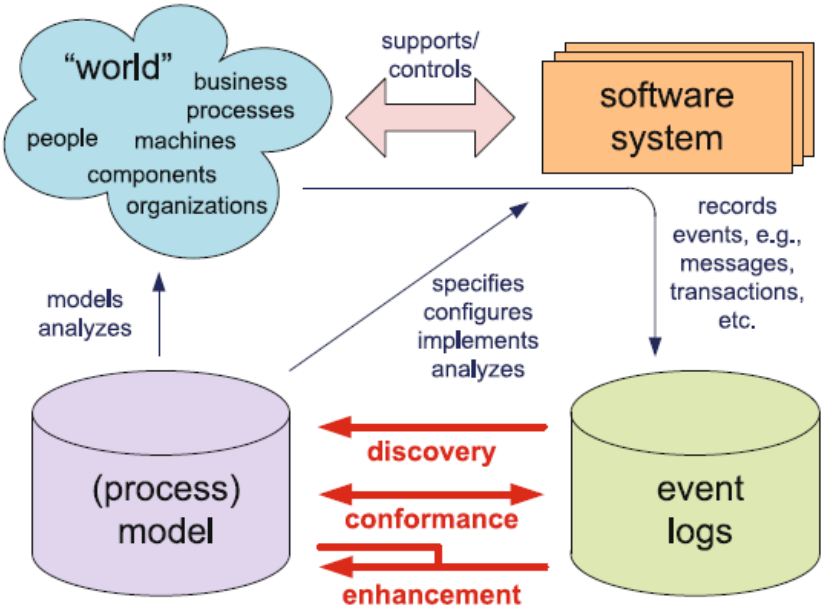

Figura 4. I tre principali tipi di Process Mining: Discovery, Conformance e Enhancement.

3. Process Enhancement: si basa sull'idea di ampliare o migliorare il modello del processo esistente, usando le informazioni raccolte nei log di eventi riguardo il processo nella sua configurazione attuale. Sebbene sia la parte di conformance checking a misurare l'allineamento fra il modello e la situazione reale, questa terza tipologia di process mining mira a modificare o estendere il modello aprioristico formulato inizialmente. Un tipico miglioramento è la riparazione, intesa come una correzione del modello volta a rappresentare in maniera più fedele la realtà. Se ad esempio due attività sono modellate in modo sequenziale ma nella realtà sono eseguite in ordine casuale, è necessario rettificare il modello in modo che rifletta correttamente la reale successione delle attività. Un altro miglioramento attuabile è l'estensione, ossia l'aggiunta al modello di una nuova prospettiva mediante la correlazione incrociata con il log. Un esempio di estensione è costituito dall'integrazione delle misure di performance nel modello di processo: servendosi dei riferimenti temporali contenuti all'interno dei log, si possono mettere in evidenza grandezze quali i colli di bottiglia, i throughput e i livelli di servizio. Allo stesso modo, si possono aggiungere informazioni

riguardo le risorse, la qualità delle metriche adottate, le regole osservate nei processi decisionali, ecc.

In modo indipendente dalle tre tipologie appena definite, van der Aalst identifica diverse prospettive[10] con le quali condurre il process mining:

- La prospettiva del control-flow si concentra sul flusso di controllo, ossia sull'ordinamento delle attività. L'obiettivo è trovare una buona rappresentazione per tutti i possibili cammini (cioè sequenze di attività), esprimendoli secondo una precisa notazione quale ad esempio la rete di Petri.
- La prospettiva organizzativa guarda alle informazioni nascoste all'interno dei log relativamente alle risorse, in particolare gli attori coinvolti (nello specifico persone, ruoli, sistemi e dipartimenti) nel processo produttivo e il tipo di relazione che intercorre fra di loro. Il fine consiste nella strutturazione dell'organizzazione, classificando ogni persona secondo il ruolo ricoperto e secondo l'unità organizzativa a cui è destinata; in alternativa, si può rappresentare la rete sociale delle persone facenti parte dell'organizzazione.
- La prospettiva del caso si focalizza sulle proprietà che distinguono ogni singolo caso. Ovviamente ciascun caso è caratterizzato dal suo cammino all'interno del processo, ma anche da chi lo ha generato e lavora su di esso. I casi sono inoltre descritti dai valori dei loro corrispondenti data elements. Ipotizzando che un caso sia rappresentato da un ordine di rifornimento, i data elements che suscitano interesse sono ad esempio il fornitore e il numero di prodotti ordinati.
- La prospettiva temporale si interessa della tempistica e della frequenza con le quali occorrono gli eventi. Quando gli eventi registrati nei log portano con sé le rispettive marche temporali (data e orario di registrazione) è possibile determinare i colli di bottiglia, misurare i livelli di servizio, tenere sotto controllo l'utilizzo delle risorse e prevedere i tempi di processo rimanenti dei casi in esecuzione.

È bene sottolineare che tali prospettive non sono pienamente esaustive e, in parte, si sovrappongono l'un con l'altra. Ciononostante, sono in grado di descrivere adeguatamente gli aspetti che il process mining punta ad analizzare. Nella gran parte dei casi si assume che il process mining sia svolto in modalità offline, ossia i processi sono analizzati in un tempo successivo per cercare di capire come possono essere migliorati o maggiormente compresi. Tuttavia, sempre più tecniche di process mining presentano la possibilità di essere applicate anche in configurazione online, ossia contemporaneamente all'esecuzione del processo. Il termine di riferimento che descrive questo aspetto è supporto operativo. Un esempio esplicativo riguarda l'individuazione di una non conformità contestualmente al manifestarsi di una deviazione. Un altro esempio fa riferimento alla previsione temporale per i casi in esecuzione in un determinato momento: dato un caso parzialmente eseguito, il

[10] W.M.P. van der Aalst, op. cit., p. 11.

restante tempo di processo è stimato sulla base delle informazioni storiche riguardanti casi simili a quello considerato. Questo significa che lo spettro di competenza del process mining è piuttosto ampio; non a caso, le tecniche di process mining attualmente disponibili non si limitano al process discovery ma sono in grado di supportare l'intero ciclo di vita BPM.

Uno degli aspetti fondamentali del process mining è l'enfasi che si pone nello stabilire una forte relazione fra il modello di processo e la realtà contenuta nei log. Per descrivere il tipo di tale relazione si utilizzano i termini play-out, play-in e replay[11].

Il play-out si riferisce al classico uso che si fa dei modelli di processo. Data una rete di Petri, è possibile generare un comportamento. La relazione play-out può essere usata sia per l'analisi sia per la fase di enactment dei processi. Un workflow engine, ossia un software che si occupa di gestire i processi aziendali, può essere visto come un "play-out engine" che controlla i vari casi e permette di compiere solo i passi consentiti dalle specifiche contenute nel modello. Anche la simulazione utilizza un play-out engine per condurre gli esperimenti; l'idea di base consiste nell'eseguire ripetutamente un modello, allo scopo di raccogliere statistiche e intervalli di fiducia. Il motore di simulazione è simile al workflow engine; la differenza principale è inerente all'ambiente d'interazione. Nella fattispecie, il motore di simulazione interagisce con un ambiente modellato mentre il workflow engine interagisce con l'ambiente reale.

Il play-in è l'opposto del play-out: l'input è costituito da un comportamento mentre l'obiettivo riguarda la costruzione di un modello. Spesso si fa riferimento al play-in come inferenza. L'algoritmo α e gli altri approcci del process discovery sono tutti esempi delle tecniche di play-in. Gran parte delle tecniche per il data mining utilizzano il play-in, ossia apprendono un modello basandosi su degli esempi. Alcuni tipici esempi di modelli sono gli alberi di decisione e le regole di associazione. Tuttavia, il data mining non si occupa di modelli di processo e pertanto le sue tecniche non sono applicabili in questo contesto. Solo di recente le tecniche di process mining, con le quali si possono scoprire modelli di processo basati sui log di eventi, sono diventate facilmente disponibili.

Il replay, invece, usa come input una coppia formata da un log di eventi e da un modello di processo. Il log è ripetuto in tempi diversi, ossia si ripercorre nel modello lo stesso cammino in più occasioni. Secondo van der Aalst, sono vari gli scopi per i quali un log di eventi può essere ripetuto[12]:

- Conformance checking: ripetendo il log si possono rilevare e quantificare le discrepanze fra il log stesso e il modello.
- Estendere il modello con frequenze e informazioni temporali: con la ripetizione del log si può vedere quali parti del modello sono percorse più frequentemente. La ripetizione può anche essere utile a individuare i colli di bottiglia del processo.

[11] W.M.P. van der Aalst, op. cit., pp. 18-19.
[12] W.M.P. van der Aalst, op. cit., pp. 19-20.

- Costruire modelli predittivi: si possono costruire modelli predittivi in seguito alla ripetizione dei log, ossia si possono realizzare previsioni per i diversi stati assunti dal modello.
- Supporto operativo: la ripetizione non si limita ai dati storici ma può comprendere anche tracce parziali di casi ancora in esecuzione, consentendo pertanto il rilevamento di deviazioni in tempo reale. Di conseguenza, è possibile lanciare un segnale di allerta prima che il caso in esame sia completato. Allo stesso modo, si può predire il tempo di processo rimanente e la probabilità di rifiuto di un caso in esecuzione, cioè non ancora completato. Tali previsioni sono utili a suggerire i passi più adatti da compiere per l'avanzamento del caso in esame.

In definitiva, il process mining rappresenta il ponte che collega il data mining con la modellazione e l'analisi dei processi. Ormai, grazie ai progressi registrati nel corso degli anni, è sempre più consueto implementare algoritmi di process mining all'interno di vari sistemi commerciali e accademici. Tuttora, il process mining è uno dei temi caldi in materia di business process management e desta l'interesse dei centri di ricerca, delle università e delle organizzazioni a livello mondiale.

Con il presente elaborato di tesi si vuole portare all'attenzione l'innovativa tecnica di analisi dei processi nota come process mining, una disciplina sorta di recente e che consente un nuovo approccio alla gestione dei processi. La sua implementazione all'interno dei sistemi aziendali permetterà alle organizzazioni di comprendere in modo più approfondito e attinente alla realtà i processi operativi, portando notevoli miglioramenti ai loro business. L'obiettivo che il presente lavoro di tesi si prefigge di raggiungere è rendere noto al lettore su quali basi si fonda il process mining, analizzandone le tecniche e le metodologie attualmente applicate ed evidenziandone i limiti. Inoltre, si vuole sensibilizzare l'opinione del lettore riguardo le prospettive di sviluppo e cambiamento che il process mining sarà in grado di portare in ambito gestionale. L'elaborato si compone di tre capitoli. Nel primo capitolo saranno illustrate le attuali tecniche di process discovery, evidenziandone pregi e limiti. Nel secondo capitolo sarà affrontato il tema relativo al conformance checking, analizzando e quantificando l'allineamento e la conformità fra il modello di processo prescelto e il log di eventi analizzato. Il terzo capitolo sarà dedicato al supporto operativo ai processi, possibile con l'esecuzione delle tecniche di process mining in modalità online. Con il supporto operativo è possibile rilevare istantaneamente le deviazioni dai comportamenti attesi, potendo così intervenire tempestivamente in modo da migliorare e affinare i processi. Infine, saranno menzionate alcune delle sfide che il process mining si troverà ad affrontare in futuro.

1. Process Discovery

Le importanti innovazioni a livello tecnologico e comunicativo avvenute negli ultimi decenni hanno prodotto radicali cambiamenti nella conduzione di un business e nell'organizzazione del lavoro. Come risultato, i processi aziendali presentano una maggiore complessità e dipendono strettamente dai sistemi informativi. Ne deriva che la modellazione dei processi sia oggi ampiamente diffusa e considerata di estrema importanza nella guida di un'organizzazione. L'operations management, un ramo delle scienze gestionali, ha nella modellazione il suo elemento fondamentale. I modelli sono usati per prendere decisioni riguardo i processi, sia in termini operativi (es. pianificazione e controllo) che strategici (es. riprogettazione); l'operations management, invece, prevede l'uso di modelli realizzati su misura, personalizzati in direzione di una determinata tecnica di analisi o destinati a risolvere un problema specifico. Al contrario, nel BPM solitamente i modelli di processo perseguono molteplici scopi: tramite la notazione BPMN, un modello può essere usato per predire risultati mediante la simulazione, discutere riguardo le responsabilità di ciascun attore e verificare che ci sia il rispetto delle procedure aziendali. Il professor van der Aalst sostiene che progettare un buon modello sia "an art rather than a science"[13], un aspetto che secondo lui accomuna il BPM e l'operations management. Pertanto, creare modelli è un compito spesso soggetto a errori, di cui i principali sono elencati di seguito[14]:

- Il modello descrive una versione idealizzata della realtà. Durante la fase di modellazione dei processi, il progettista tende a concentrarsi sui comportamenti normali o auspicati. Ad esempio, il modello realizzato potrebbe spiegare solo l'80% dei casi possibili e, nonostante ciò, essere considerato rappresentativo. Tipicamente questa approssimazione non è valida; basti pensare che il 20% dei casi non considerati potrebbe rappresentare la causa della maggior parte dei problemi riscontrati. Un'eccessiva semplificazione del modello è dovuta a diverse ragioni. Il progettista e il management dell'azienda potrebbero non essere consapevoli delle deviazioni che avvengono nella realtà, per cui il modello non ne tiene conto. Inoltre, la percezione delle persone è distorta e influenzata a seconda del ruolo ricoperto all'interno dell'organizzazione, il cui esito consiste in un modello privo di obiettività. I modelli realizzati a mano sono inclini alla soggettività e spesso presentano una struttura semplificata e poco approfondita per agevolarne la comprensibilità, a scapito di un adeguato allineamento con la realtà.
- L'incapacità di catturare in modo adeguato il comportamento umano. Sebbene i modelli matematici più semplici siano sufficienti a modellare macchinari o persone operanti lungo una linea produttiva, non sono in grado di rappresentare adeguatamente il comportamento delle persone

[13] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, p. 30. ["un'arte piuttosto che una scienza"].
[14] W.M.P. van der Aalst, op. cit, pp. 30-31.

coinvolte in più di un processo. Un lavoratore avente responsabilità su diversi processi deve distribuire la sua attenzione su ognuno di essi, per cui è difficile modellare un singolo processo in modo indipendente; inoltre l'essere umano, durante la sua attività lavorativa, non mantiene la medesima velocità. Nella maggioranza dei processi, è facile rilevare che le persone impiegano più tempo a completare un compito e lavorano per un numero effettivo di ore più basso se non hanno molte mansioni da svolgere. Ciononostante, molti modelli di simulazione estraggono i tempi di servizio da una stessa distribuzione di probabilità e utilizzano finestre temporali fisse per la disponibilità delle risorse.

- Il modello è ad un livello di astrazione errato. In base ai dati in ingresso e alle esigenze a cui si vuole rispondere, è necessario scegliere un adeguato livello di astrazione. Un buon modello presenta il giusto equilibrio fra il livello di astrazione e il livello di dettaglio. Se il modello fosse troppo astratto non potrebbe fornire una risposta attendibile ai problemi e alle questioni più rilevanti; se, invece, fosse eccessivamente dettagliato risulterebbe di difficile comprensione e genererebbe complicazioni nel reperimento dei dati di input necessari. Sin dal principio, è bene accertarsi di aver optato per un livello di astrazione che sia adatto al processo in esame, in quanto eventuali modifiche richiederebbero un considerevole dispendio di tempo. Sfortunatamente, problemi e difficoltà possono sorgere a prescindere dal livello scelto.

Come detto in precedenza, gli errori sopracitati sono solo alcune delle imprecisioni che si commettono in fase di realizzazione dei modelli a mano. Un modello non adeguato può condurre a conclusioni errate; solo analisti e progettisti di elevata esperienza sono in possesso di capacità e di conoscenze tali da creare modelli attendibili e predittivi, i quali possono essere usati come elemento basilare in fase di implementazione o di riprogettazione. A valle di tali considerazioni, è pertanto raccomandabile ricorrere all'uso dei dati presenti all'interno dei sistemi informativi aziendali; in ottemperanza a tale raccomandazione, il process mining permette di progettare modelli che abbiano come loro fondamento essenziale gli eventi realmente occorsi. Inoltre, il process mining non persegue l'obiettivo della creazione di un singolo modello, ma bensì punta a fornire varie panoramiche sulla medesima situazione reale, a diversi livelli di astrazione. Ne deriva che non è semplice progettare modelli in modo corretto; tuttavia, essi sono di estrema importanza al fine di gestire i processi in modo diligente e rigoroso.

Costruiti i modelli di processo, si procede studiandoli e controllandoli in maniera minuziosa. Tipicamente, sono due gli approcci convenzionali con i quali condurre un'analisi, basata sui modelli di processo, approfondita e veritiera: la verifica e l'analisi delle prestazioni. La verifica è volta ad accertare la correttezza del processo o, più in generale, del sistema in esame. L'analisi delle prestazioni si concentra su varie grandezze e può essere definita secondo diverse modalità. In particolare, il professor van der Aalst identifica tre dimensioni di performance a cui guardare con interesse e attenzione:

tempo, costo e qualità[15]. Per ciascuna delle tre dimensioni si possono definire differenti Key Performance Indicators (KPI). Partendo dalla dimensione tempo, van der Aalst elenca alcuni esempi di indici prestazionali[16], di seguito riportati:

- Il lead time è il tempo complessivo che intercorre fra la creazione di un caso ed il suo completamento. Può essere espresso anche come valore medio dei lead time di tutti i casi registrati; tuttavia, non bisogna trascurare la varianza e la sua composizione. È differente la valutazione fra casi con lead time più o meno simili e casi fra i quali alcuni presentano lead time, ad esempio, di poche ore e altri di molte settimane. Il livello di servizio può essere espresso come percentuale di casi aventi lead time inferiori a un valore di soglia prefissato.
- Il tempo di servizio è il tempo attualmente impiegato per la lavorazione di un caso. Può essere misurato per singola attività o per intero caso. Considerando l'ipotesi di casi concomitanti, ossia contemporaneamente in esecuzione, il tempo di servizio totale (somma dei tempi trascorsi su ciascuna attività) potrebbe rivelarsi maggiore del lead time; ciononostante, solitamente il tempo di servizio rappresenta solo una frazione del lead time.
- Il tempo di attesa è il tempo durante il quale un caso attende la disponibilità di una risorsa per poter essere processato. Come per il tempo di servizio, anche il tempo di attesa può essere misurato per singola attività o per intero caso. Un esempio esplicativo è rappresentato dal pronto soccorso, dove i pazienti attendono per un certo tempo prima di essere visitati.
- Il tempo di sincronizzazione è il tempo in cui un'attività non è ancora in esecuzione perché si trova in attesa di un segnale esterno o del completamento di un ramo parallelo del processo.

Per quanto riguarda la dimensione costo, l'Activity Based Costing (ABC) è uno dei principali modelli utilizzati per svolgere l'analisi. Il costo di un'attività può dipendere dal tipo di risorsa adoperata, dal suo utilizzo o dalla durata dell'attività stessa. Uno dei KPI presenti nella gran parte dei processi è l'utilizzo medio delle risorse lungo un arco temporale (es. la sala operatoria di un ospedale è stata utilizzata per l'85% del tempo nel corso degli ultimi due mesi). La qualità, invece, è la dimensione di analisi che guarda al prodotto o servizio da consegnare al cliente. Può essere misurata in vari modi: alcuni esempi sono la customer satisfaction, valutata tramite questionari, e il numero dei prodotti difettosi. In definitiva, la verifica si focalizza sulla correttezza logica del processo modellato mentre l'analisi delle prestazioni punta a migliorare i processi rispetto alle tre dimensioni tempo, costo e qualità. Occorre aggiungere che i modelli analitici spesso richiedono molte assunzioni in fase di formulazione e possono essere applicati solo per rispondere a particolari problemi. Per via di tale considerazione, si rende necessaria un'ulteriore tecnica di analisi: la simulazione. Molti strumenti per il BPM contengono al loro interno un software di simulazione, con il quale è possibile sperimentare scenari diversi e ottenere le

[15] W.M.P. van der Aalst, op. cit., p. 55.
[16] W.M.P. van der Aalst, op. cit., pp. 55-56.

relative misure di performance. Sebbene molte organizzazioni abbiano provato ad usare la simulazione per l'analisi dei loro processi operativi, poche sono coloro che utilizzano la simulazione in maniera strutturata e, soprattutto, efficace[17]. Le cause sono da ricondurre alla mancanza di formazione e ai limiti che gli attuali strumenti di simulazione presentano. Inoltre, i modelli di simulazione sono inclini a eccessive semplificazioni; nella fattispecie, il comportamento delle risorse è modellato in modo approssimativo. Come già anticipato, le persone non lavorano con velocità costante e distribuiscono la loro attenzione su svariati processi. La tendenza delle aziende moderne è di tenere traccia degli eventi registrandoli sotto forma di log; alcune di esse possono avvalersi di modelli di processo notevolmente accurati che conservano nei propri sistemi BPM.

Tornando all'analisi dei processi basata sui modelli, condotta tramite la verifica e l'analisi delle prestazioni, è facile comprendere che presenta un ragguardevole limite[18]: si basa su modelli di alta qualità. Ne consegue che non avrebbe alcun senso eseguire questo tipo di analisi nel caso in cui non ci fosse un buon allineamento fra il modello e la situazione reale. Un modello di processo disallineato descrive una versione idealizzata che lo rende pressoché inutile considerando i vari tipi di deviazioni che si manifestano nella realtà. Lo stesso si può dire dei modelli di simulazione, data la loro natura imperfetta. Il process mining affronta questi problemi cercando di mettere in relazione i modelli con i dati riguardanti i processi.

L'analisi e la modellazione dei processi è uno dei due pilastri sui quali si erge il process mining; il secondo è rappresentato dal data mining. Sebbene tale materia non sia oggetto di studio del presente elaborato, è bene spiegarne le nozioni principali allo scopo di facilitare la comprensione del process mining e delle sue tecniche applicative. il data mining è definito come "the analysis of (often large) data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner"[19]. Solitamente, i dati in ingresso sono raccolti e rappresentati in forma tabellare; diversamente, in uscita possono presentarsi varie tipologie quali grafici, equazioni, regole, strutture ad albero[20], ecc.

Tabella 1.1 Dati di circa 860 persone decedute utili a studiare gli effetti di alcol, fumo e peso corporeo sull'aspettativa di vita.

Bevitore	Fumatore	Peso (kg)	Età
Sì	Sì	120	44
No	No	70	96
Sì	No	72	88
…	…	…	…

[17] W.M.P. van der Aalst, op. cit., p. 56.

[18] W.M.P. van der Aalst, op. cit., p. 57.

[19] D. Hand, H. Mannila, and P. Smyth. Principles of Data Mining, MIT Press, Cambridge, MA, 2001. ["l'analisi di insiemi di dati (spesso molto ampi) volte a trovare relazioni insospettate e a sintetizzare i dati con modalità innovative che siano utili e comprensibili al proprietario di tali dati"].

[20] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, p. 59.

In Tabella 1.1 è raffigurato un esempio di dati di ingresso, preso e riportato dall'opera del professor van der Aalst[21]. Tali dati sono utilizzati come input per l'applicazione di algoritmi, con i quali condurre un'analisi di data mining. In Tabella 1.1 è stata selezionata a scopo illustrativo solo una piccola parte degli 860 individui del campione. La colonna 'Bevitore' indica se la persona fosse solita assumere alcol. La colonna 'Fumatore' indica se la persona fosse un fumatore o meno. La colonna 'Peso' indica il peso corporeo della persona deceduta. La colonna 'Età' indica l'età alla quale è avvenuto il decesso della persona. Ogni riga rappresenta una persona diversa. Solitamente si fa riferimento a ciascuna riga con il termine istanza; in alternativa, si possono utilizzare i termini individuo, entità, caso, oggetto e registrazione. Per riferirsi a una singola colonna, invece, si usa il termine variabile. Le variabili possono essere chiamate con vari vocaboli: attributi, caratteristiche e data elements sono i più comuni. Nell'esempio di Tabella 1.1 è quindi presente un insieme di dati (o data set) relativi a quattro variabili. È inoltre possibile distinguere fra variabili categoriche (o di categoria) e variabili numeriche[22]; le prime indicano l'appartenenza ad una certa categoria e possono assumere un numero limitato di valori, mentre le seconde esprimono un numero e pertanto possono essere ordinate. Le variabili categoriche sono ulteriormente suddivise in variabili nominali e variabili ordinali. Le variabili nominali non presentano un ordinamento logico; alcuni esempi sono una variabile booleana (es. vero o falso), i colori (es. rosso, blu, giallo, ecc.) e le nazioni (es. Italia, Francia, Germania, ecc.). Le variabili ordinali, al contrario, possono essere ordinate secondo una precisa logica, che dipende dal tipo di dati in questione. Ad esempio, i risultati di un esame sostenuto da un certo numero di studenti possono essere ordinati con i valori "promosso con lode", "promosso" e "bocciato". Prima di attuare una qualsiasi tecnica di data mining, si procede con una selezione dei dati, eliminando le variabili ritenute poco rilevanti e le istanze contaminate da fattori estranei ai fini dell'analisi. Facendo un rapido parallelo, si nota che il process mining introduce due nozioni, gli eventi e i casi (ognuno costituito da eventi), contro la sola nozione di istanza su cui opera il data mining. In aggiunta, gli eventi sono cronologicamente ordinati, mentre l'ordinamento delle istanze, come si può osservare nell'esempio in Tabella 1.1, non ha alcun significato. In presenza di particolari problemi, è possibile convertire un log di eventi in un più semplice data set, con il quale eseguire il data mining. A quest'ultima procedura, si fa riferimento con il termine feature extraction[23], ossia l'estrazione di caratteristiche, la quale consente una riduzione delle dimensioni del problema. Per quanto riguarda le tecniche di data mining, la classificazione che adopera il professor van der Aalst prevede due famiglie preminenti: l'apprendimento supervisionato e l'apprendimento non supervisionato[24]. La prima tipologia assume che i dati siano contrassegnati: in sostanza, si sceglie una variabile di risposta con la quale

[21] W.M.P. van der Aalst, op. cit., p. 60.
[22] W.M.P. van der Aalst, op. cit., p. 61.
[23] W.M.P. van der Aalst, op. cit., p. 62.
[24] W.M.P. van der Aalst, op. cit., pp. 62-63.

contrassegnare ogni singola istanza. Riprendendo l'esempio precedente, ogni studente potrebbe essere etichettato con "promosso con lode", "promosso" e "bocciato". Le variabili rimanenti sono tutte definite come variabili predittive. Spesso, soprattutto in ambito statistico, la variabile di risposta è nota come variabile dipendente e le variabili predittive sono chiamate variabili indipendenti. L'obiettivo è spiegare il comportamento della variabile dipendente rispetto alle variabili indipendenti, cercando di cogliere le eventuali relazioni intercorrenti fra l'una e le altre. A seconda del tipo di variabile dipendente, le tecniche per l'apprendimento supervisionato si ripartiscono in due sottogruppi: classificazione e regressione. Le tecniche di classificazione prevedono l'assunzione di una variabile di risposta di tipo categorico e perseguono come scopo la classificazione delle istanze sulla base delle variabili indipendenti. Tramite l'uso della classificazione si possono realizzare costrutti fra i quali i più diffusi sono gli alberi di decisione. Contrariamente alla classificazione, le tecniche di regressione richiedono una variabile dipendente di tipo numerico; il loro obiettivo è trovare una funzione che descriva adeguatamente e con il minimo errore la dipendenza fra variabile di risposta e variabili predittive. La tecnica di regressione usata più assiduamente è la regressione lineare. L'apprendimento non supervisionato, all'opposto, non contrassegna i dati, ossia non determina quale sia la variabile dipendente[25]. Il professor van der Aalst considera due tipi di apprendimento non supervisionato: l'analisi dei gruppi (o clustering) e il pattern discovery. Gli algoritmi di clustering esaminano i dati al fine di segnalare gruppi di istanze fra loro simili. Diversamente dalla classificazione, l'attenzione non è posta sulla variabile di risposta ma bensì sull'istanza; per esempio, l'obiettivo potrebbe essere trovare gruppi omogenei di persone decedute (Tabella 1.1). Le tecniche maggiormente note sono il k-means clustering e l'agglomerative hierarchical clustering. L'obiettivo del pattern discovery è scoprire quali sono i cammini percorsi all'interno dei dati, cercando di trovare regole di tipo *IF X THEN Y* che li descrivano, dove *X* e *Y* rappresentano i valori delle diverse variabili. Ad esempio, per la Tabella 1.1 potrebbe valere la regola *IF fumatore = no AND età ≥ 70 THEN bevitore = sì*. Per questa tipologia di apprendimento non supervisionato, la tecnica più conosciuta è rappresentata dalle regole di associazione.

Ora che, al fine di facilitare la compressione del process mining, i due fondamenti sono stati introdotti, si può concentrare l'attenzione sulla prima tipologia di tale disciplina: il process discovery. Non è possibile adoperare il process mining in assenza di log di eventi corretti e appropriati[26]; il principio su cui si incentra il process discovery consiste nella scoperta o nella deduzione di un modello di processo, partendo da log di eventi. Per cui si analizzano gli eventi con un prospettiva orientata al processo. In Figura 1.1 è mostrato il flusso di lavoro complessivo del process mining, ponendo l'enfasi sul ruolo ricoperto dai dati. Inizialmente si hanno a disposizione dati non ancora lavorati, le cui origini sono di vari formati. Una fonte di dati può essere un semplice file come, ad esempio, un foglio excel o una

[25] W.M.P. van der Aalst, op. cit., p. 64.
[26] W.M.P. van der Aalst, op. cit., p. 95.

tabella proveniente da un database. In ogni caso, raramente i dati sono interamente contenuti all'interno di una singola fonte; più realisticamente, i dati sono disseminati su fonti differenti e spesso sono necessari notevoli sforzi per raccoglierne i più rilevanti.

In ambito di Business Intelligence (BI), la frase "Extract, Transform, and Load" (ETL) sintetizza nel miglior modo possibile il processo[27] di seguito descritto: l'estrazione dei dati dalle rispettive fonti, la loro trasformazione al fine di adattarli alle esigenze operative dell'organizzazione e il loro caricamento all'interno del sistema in questione (es. data warehouse). Nello specifico, un data warehouse è il magazzino informatico nel quale sono conservati tutti i dati operativi e transazionali di un'organizzazione. Il data warehouse non produce dati ma semplicemente li preleva dai sistemi operativi. Lo scopo è di conferire univocità alle informazioni aziendali, in modo che possano essere utilizzate per eseguire analisi, previsioni, reportistiche, ecc. Come è mostrato in Figura 1.1, le attività ETL sono utili a popolare il data warehouse. Nel caso in cui il data warehouse esista, molto probabilmente conterrà dati utilizzabili come input per il process mining; tuttavia, molte organizzazioni non sono in possesso di data warehouse appropriati, considerato che in essi sono conservate solo informazioni relative all'end-to-end process mining, ossia dati relativi solamente ai clienti. A prescindere dalla presenza di un data warehouse, è indispensabile estrarre i dati e convertirli in log di eventi. Per tale fine, la fase di scoping (o perlustrazione dei dati) è di estrema importanza; infatti, spesso il problema non riguarda la conversione ma la selezione dei dati adeguati. Dopo aver completato la conversione, il log di eventi ottenuto è sottoposto a filtraggio[28]. Al momento della conversione da dati a log di eventi si opera una selezione grossolana (coarse-grained scoping), che fornisce determinati risultati; il filtraggio, che è un processo iterativo, si basa su tali risultati e agisce in

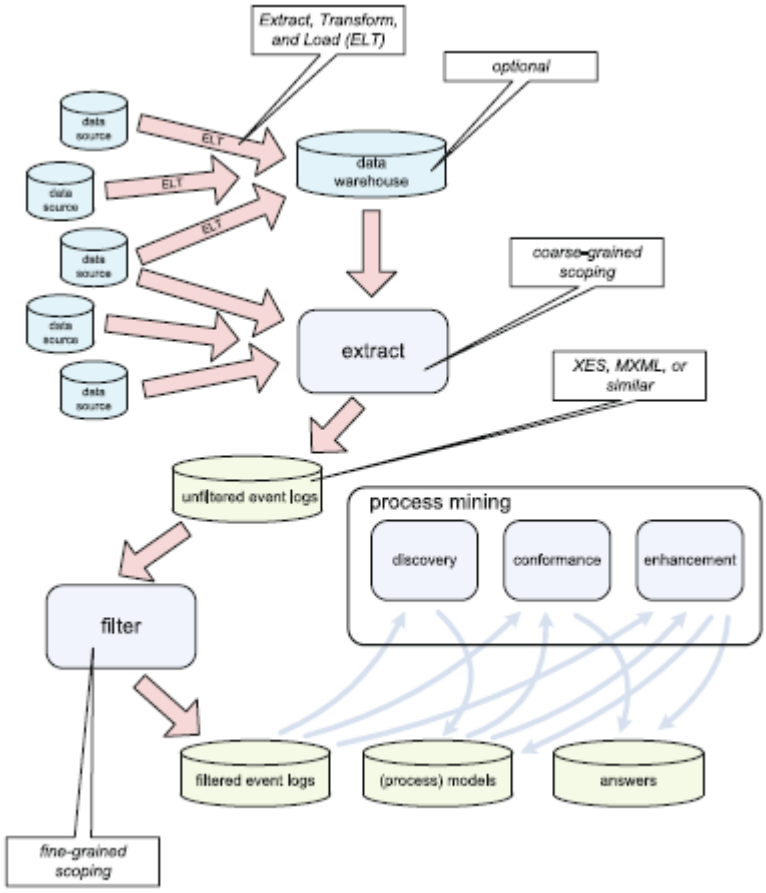

Figura 5.1 Flusso di lavoro del process mining partendo da fonti di dati diverse.

[27] W.M.P. van der Aalst, op. cit., p. 97.
[28] W.M.P. van der Aalst, op. cit., p. 98.

maniera più approfondita e dettagliata (fine-grained scoping). Tipicamente, sono necessarie diverse iterazioni di estrazione e filtraggio per ottenere dei buoni risultati; i log filtrati che si ricavano sono la base di partenza delle tre tipologie di process mining già menzionate in fase di introduzione.

Riguardo il log di eventi, in Tabella 1.2 sono mostrati alcuni esempi utili a chiarirne il concetto, riportati dall'opera del professor van der Aalst[29]. In tale tabella sono illustrate le classiche informazioni presenti

Tabella 1.2 Esempi di log di eventi riferiti alla richiesta di risarcimento.

ID Caso	ID Evento	Proprietà			
		Timestamp	Attività	Risorsa	Costo
1	35654423	30/12/2010 11:02	Register Request	Pete	50
	35654424	31/12/2010 10:06	Examine thoroughly	Sue	400
	35654425	05/01/2011 15:12	Check ticket	Mike	100
	35654426	06/01/2011 11:18	Decide	Sara	200
	35654427	07/01/2011 14:24	Reject request	Pete	200
2	35654483	30/12/2010 11:32	Register Request	Mike	50
	35654485	30/12/2010 12:12	Check ticket	Mike	100
	35654487	30/12/2010 14:16	Examine casually	Pete	400
	35654488	05/01/2011 11:22	Decide	Sara	200
	35654489	08/01/2011 12:05	Pay compensation	Ellen	200

all'interno di un log di eventi fruibile dal process mining. L'assunzione di base consiste nel considerare ciascun log di eventi come il contenitore di dati relativi ad un solo processo. Inoltre, ogni evento del log si riferisce a un'istanza del singolo processo in esame, la quale di frequente è denominata con il termine caso. In Tabella 1.2 vi sono due casi ognuno composto da cinque eventi. Spesso gli eventi sono relativi ad alcune attività, dalle quali prendono il nome; una sequenza finita di eventi definisce una traccia[30] o cammino. Sono presenti anche informazioni aggiuntive quali la risorsa, il costo associato all'evento e le coordinate temporali (colonna Timestamp); quest'ultima in particolare favorisce le analisi di performance del processo, come, considerando gli esempi di tabella 1.2, il tempo di attesa del paziente fra la fine di un'attività e l'inizio della successiva. Per tali proprietà si usa l'appellativo di attributi. Occorre ricordare l'importanza di una stretta relazione fra eventi e casi al fine di attuare correttamente le tecniche di process mining. Come già detto, il process discovery si propone di costruire modelli di processo sulla base dei comportamenti che si manifestano nei log; il professor van der Aalst fornisce la seguente definizione:" A process discovery algorithm is a function that maps L onto a process model such that the model is "representative" for the behavior seen in the event log.

[29] W.M.P. van der Aalst, op. cit., p. 99.
[30] W.M.P. van der Aalst, op. cit., p. 104.

The challenge is to find such an algorithm"[31]. Con L si intende indicare un generico log di eventi. La definizione è alquanto ampia e vaga, non esprime un obiettivo relativamente al formato dell'algoritmo; inoltre, un log di eventi con molteplici potenziali attributi può essere usato come input, seppur in assenza di specifiche richieste. Per rendere più concreto il concetto, si definisce come formato target la rete di Petri, mentre si utilizza come input un log di eventi semplice L. Quest'ultimo è definito come un insieme multiplo di tracce, ciascuna delle quali è composta da elementi appartenenti all'insieme A dei nomi delle attività.

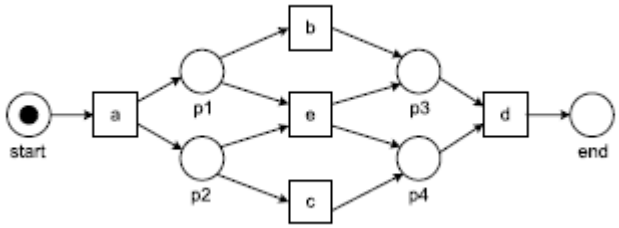

Figura 1.2 Workflow net del log di eventi L_1.

Ad esempio il log semplice $L_1 = [(a, b, c, d)^3, (a, c, b, d)^2, (a, e, d)]$ contiene 6 casi $(3 + 2 + 1)$ e $3 \times 4 + 2 \times 4 + 1 \times 3 = 23$ eventi. L'obiettivo è pertanto realizzare una rete di Petri che sia in grado di replicare quanto osservato in L_1. Pertanto il problema può essere riformulato nel seguente modo: un algoritmo di process discovery è una funzione γ che definisce un log L appartenente ad A lungo una rete di Petri marcata $\gamma(\mathrm{L}) = (N, M)$. Idealmente, N è una workflow net di tipo sound e tutte le tracce presenti in L corrispondono a possibili sequenze che soddisfano la regola di scatto. Le reti workflow (workflow nets o WF-nets) sono una sottoclasse delle reti di Petri e presentano due specifici posti (*i* e *o*)dedicati rispettivamente all'inizio e alla fine del processo. Inoltre, i nodi componenti la rete sono tutti elementi costituenti di cammini che partono da *i* e terminano in *o*. Con M si intende la marcatura con la quale i token si dispongono sulla rete. Una WF-net rispetta la condizione di soundness[32], ossia può essere considerata valida e solida, solamente se:

- $(N, [i])$ è sicura, cioè i posti della rete non contengono molteplici token in contemporanea.
- Per qualsiasi marcatura $M \in [N, [i]), o \in M$ implica $M = o$, ossia se il posto finale è marcato tutti gli altri posti dovrebbero essere vuoti (proper completion).
- Per qualsiasi marcatura $M \in [N, [i]), [o] \in [N, M)$, cioè è sempre possibile marcare il posto finale (option to complete).
- $(N, [i])$ non contiene transizioni morte, ossia tutte le parti del modello sono potenzialmente raggiungibili.

[31] W.M.P. van der Aalst, op. cit., p. 125. ["Un algoritmo di process discovery è una funzione che definisce L all'interno di un modello di processo in modo tale che il modello sia "rappresentativo " del comportamento osservato nel log. La sfida consiste nel trovare un tale algoritmo"].
[32] W.M.P. van der Aalst, op. cit., p. 127.

Affinché l'algoritmo di process discovery sia rappresentativo, è necessario che presenti il corretto trade-off fra i seguenti criteri di qualità:

- Fitness: il modello di process discovery dovrebbe consentire la replicazione del comportamento osservato nel log.
- Precisione: il modello di process discovery non dovrebbe permettere comportamenti che non presentino alcuna relazione con quanto osservato nel log.
- Generalizzazione: il modello di process discovery dovrebbe generalizzare il comportamento esemplificativo osservato nel log.
- Semplicità: il modello di process discovery dovrebbe essere il più semplice possibile.

Un modello con un buon fitness è in grado di replicare la maggior parte delle tracce presenti nel log. La precisone è riferita al concetto di underfitting; un modello poco preciso soffre di underfitting, ossia consente anche comportamenti diversi da quelli osservati nel log di eventi in esame. La generalizzazione è relativa alla nozione di overfitting. Un modello in overfitting non è abile a generalizzare in maniera sufficiente, pertanto è troppo specifico e fa un eccessivo riferimento agli esempi osservati nel log. La qualità di semplicità prende spunto dal principio del Rasoio di Occam: è inutile aumentare, oltre la quantità sufficiente, il numero di entità e ipotesi necessarie a spiegare un particolare fenomeno. Seguendo questo pensiero, si ricerca il più semplice modello di process discovery che possa spiegare i comportamenti che si sono manifestati nel log di eventi.

Un semplice modello per il process discovery è rappresentato dall'algoritmo α; è un esempio di funzione γ, secondo la definizione precedentemente fornita. Partendo da un log di eventi semplice, tale algoritmo produce una rete di Petri che, auspicabilmente, possa riprodurre in maniera adeguata il log; pertanto, è a tutti gli effetti una tecnica di tipo play-in. L'input per l'algoritmo α è rappresentato da un log di eventi semplice L appartenente all'insieme delle attività A, le quali corrispondono alle transizioni nella rete di Petri ricavata. L'output dell'algoritmo α consiste in una rete di Petri marcata $\propto (L) = (N, M)$. L'obiettivo consta nella deduzione di WF-nets, per cui si può omettere la marcatura iniziale e scrivere $\propto (L) = N$; è implicito che la marcatura di partenza sia $M = [i]$. L'algoritmo α scansiona il log lungo determinati cammini. Ad esempio, se l'attività a è seguita dall'attivita b ma b non è mai succeduta da a, si può assumere che ci sia una dipendenza causale fra a e b. Per riflettere tale relazione, la rete di Petri dovrebbe mostrare un posto che colleghi a con b. Per individuare i cammini più rilevanti, si utilizzano quattro diverse relazioni di ordinamento basate sui log[33]:

- $a >_L b$ se e solo se è presente una traccia $\sigma = (t_1, t_2, t_3, \dots, t_n)$ e $i \in \{1, \dots, n-1\}$ tale che $\sigma \in L$ e $t_i = a$ e $t_{i+1} = b$
- $a \rightarrow_L b$ se e solo se $a >_L b$ e $b \not>_L a$
- $a \#_L b$ se e solo se $a \not>_L b$ e $b \not>_L a$

[33] W.M.P. van der Aalst, op. cit., p. 129.

- $a \,||_L b$ se e solo se $a >_L b$ e $b >_L a$

Riprendendo il log di eventi L_1, si possono rilevare le relazioni di ordinamento[34] di seguito elencate:

$>_{L_1} = \{(a,b),(a,c),(a,e),(b,c),(c,b),(b,d),(c,d),(e,d)\}$

$\rightarrow_{L_1} = \{a,b),(a,c),(a,e),(b,d),(c,d),(e,d)\}$

$\#_{L_1} = \{(a,a),(a,d),(b,b),(b,e),(c,c),(c,e),(d,a),(d,d),(e,b),(e,c),(e,e)\}$

$||_{L_1} = \{(b,c),(c,b)\}$

La relazione $>_{L_1}$ contiene le coppie di attività che sono direttamente collegate. Ad esempio $c >_{L_1} d$ indica che d segue direttamente c nella traccia (a,b,c,d), mentre $d \not>_{L_1} c$ indica che c non segue mai direttamente d in nessuna traccia del log. La relazione $\rightarrow_{L_1}$ comprende le coppie di attività aventi relazione causale; ad esempio, $c \rightarrow_{L_1} d$ significa che a volte d segue direttamente c ma non si verifica mai il percorso inverso ($c >_{L_1} d$ e $d \not>_{L_1} c$). La relazione $b \,||_{L_1} c$ vuol dire che $b >_{L_1} c$ e $c >_{L_1} b$, ossia a volte c segue b e altre volte accade il percorso contrario. La relazione $b \,\#_{L_1} e$ significa che $b \not>_{L_1} e$ e $e \not>_{L_1} b$. Ogni coppia di attività presenta una fra le relazioni $\rightarrow_{L_1}$, $\#_{L_1}$ e $||_{L_1}$. Pertanto, è possibile riassumere il footprint di un log in forma matriciale, come mostrato in Tabella 1.3.

Tabella 1.3 Footprint del log di eventi L_1.

	a	*b*	*c*	*d*	*e*
a	$\#_{L_1}$	$\rightarrow_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$
b	$\leftarrow_{L_1}$	$\#_{L_1}$	$||_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
c	$\leftarrow_{L_1}$	$||_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$
d	$\#_{L_1}$	$\leftarrow_{L_1}$	$\leftarrow_{L_1}$	$\#_{L_1}$	$\leftarrow_{L_1}$
e	$\leftarrow_{L_1}$	$\#_{L_1}$	$\#_{L_1}$	$\rightarrow_{L_1}$	$\#_{L_1}$

Dopo aver spiegato l'idea di base e mostrato alcuni esempi, si può precedere con la descrizione dall'algoritmo α[35], definito dai seguenti passi:

1. $T_L = \{t \in T \mid \exists_{\sigma \in L}\, t \in \sigma\}$
2. $T_I = \{t \in T \mid \exists_{\sigma \in L}\, t = first(\sigma)\}$
3. $T_O = \{t \in T \mid \exists_{\sigma \in L}\, t = last(\sigma)\}$
4. $X_L = \{(A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_L b \wedge \forall_{a_1 a_2 \in A}\, a_1 \#_L a_2 \wedge \forall_{b_1 b_2 \in B}\, b_1 \#_L b_2\}$
5. $Y_L = \{(A,B) \in X_L \mid \forall_{(A',B') \in X_L}\, A \subseteq A' \wedge B \subseteq B' \implies (A,B) = (A',B')\}$
6. $P_L = \{p_{(A,B)} \mid (A,B) \in Y_L\} \cup \{i_L, o_L\}$
7. $F_L = \{(a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A\} \cup \{(p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$

[34] W.M.P. van der Aalst, op. cit., p. 130.
[35] W.M.P. van der Aalst, op. cit., p. 133.

8. $\alpha(L) = (P_L, T_L, F_L)$

Il log di eventi L è definito su un insieme di attività T. Al passo 1 si controllano quali attività compaiono all'interno del log (T_L), le quali costituiranno le transizioni della WF-net generata. Al passo 2 si considera l'insieme delle attività iniziali T_I, ossia le attività che occupano la prima posizione di determinate tracce. T_O è invece l'insieme delle attività finali, cioè le attività che occupano l'ultima posizione di determinate tracce (passo 3). I passi 4 e 5 costituiscono il nucleo dell'algoritmo α; il fine consiste nella definizione dei posti della rete workflow e delle loro connessioni. Per raggiungere tale scopo si costruiscono posti chiamati $p_{(A,B)}$, tali che A sia l'insieme delle transizioni di input e B sia l'insieme delle transizioni di output. Come si può osservare in Figura 1.3, gli elementi di A presentano dipendenze causali con gli elementi di B, cioè per tutte le coppie $(a, b) \in A \times B$: $a \rightarrow_L b$. Inoltre, gli elementi di A non si susseguono l'un con l'altro, ossia per le attività $a_1, a_2 \in A$: $a_1 \#_L a_2$; tale considerazione è valida anche per B. Considerando solo le righe e le colonne relative a $A \cup B$, si ottiene il footprint in forma matriciale mostrato in Tabella 1.4. Si compone di quattro quadranti: due di questi (in alto a sinistra e in basso a destra) contengono solamente il simbolo #, perciò gli elementi di A non si succedono l'uno dopo l'altro e, allo stesso modo, gli elementi di B. Il quadrante in alto a destra contiene solo il simbolo →, per cui ciascun elemento di A può essere seguito da un elemento di B, ma non si verifica mai il contrario. Per simmetria, il quadrante in alto a sinistra contiene solo il simbolo ←.

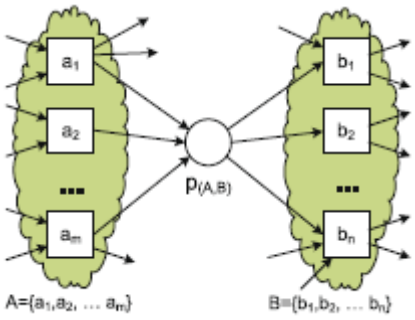

Figura 1.3 Il posto $p_{(A,B)}$ che connette le transizioni dell'insieme A con le transizioni dell'insieme B.

Tabella 1.4 Footprint corrispondente di $A=\{a_1,a_2,...,a_m\}$ e $B=\{b_1,b_2,...,b_n\}$.

	a_1	a_2	…	a_m	b_1	b_2	…	b_n
a_1	#	#	…	#	→	→	…	→
a_2	#	#	…	#	→	→	…	→
…	…	…	…	…	…	…	…	…
a_m	#	#	…	#	→	→	…	→
b_1	←	←	…	←	#	#	…	#
b_2	←	←	…	←	#	#	…	#
…	…	…	…	…	…	…	…	…
b_n	←	←	…	←	#	#	…	#

Riprendendo il log L_1, si può notare che l'insieme X_L delle coppie soddisfacenti i requisiti del passo 4 sono le seguenti:

$$X_{L_1} = \{(\{a\},\{b\}),(\{a\},\{c\}),(\{a\},\{e\}),(\{a\},\{b,e\}),(\{a\},\{c,e\}),(\{b\},\{d\}),(\{c\},\{d\}),$$
$$(\{e\},\{d\}),(\{b,e\},\{d\}),(\{c,e\},\{d\})\}$$

Nel passo 5, si rimuovono da queste le coppie non massime, ossia le coppie che non contengono tutte le attività collegate dal posto in questione. Nel caso di L_1 si ottiene:

$$Y_{L_1} = \{(\{a\},\{b,e\}),(\{a\},\{c,e\}),(\{b,e\},\{d\}),(\{c,e\},\{d\})\}$$

Ogni elemento di $(A,B) \in Y_L$ corrisponde a un posto $p_{(A,B)}$, il quale collega le transizioni dell'insieme A alle transizioni dell'insieme B. In aggiunta, P_L comprende anche un unico posto iniziale i_L e un unico posto finale o_L (passo 6). Nel passo 7, si procede con la generazione degli archi della WF-net; le transizioni comprese in T_I hanno i_L come posto di inizio, mentre le transizioni comprese in T_O hanno o_L come posto di fine. Tutti i posti $p_{(A,B)}$ presentano in ingresso nodi appartenenti all'insieme A e in uscita nodi appartenenti all'insieme B. Il risultato consiste in una rete di Petri $\alpha(L) = (P_L, T_L, F_L)$ capace di descrivere il comportamento osservato nel generico log L.

Sfruttando l'esempio che il professor van der Aalst ha riportato sulla sua opera, si può esibire una dimostrazione dell'algoritmo α[36]. Si consideri il seguente log di eventi:

$$L_5 = [(a,b,e,f)^2,(a,b,e,c,d,b,f)^3,(a,b,c,e,d,b,f)^2,(a,b,c,d,e,b,f)^4,(a,e,b,c,d,b,f)^3]$$

Tale log presenta il footprint rappresentato in Tabella 1.5, con cui è possibile applicare l'algoritmo α.

Tabella 1.5 Footprint del log di eventi L_5.

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

Assumendo $L = L_5$, si attuano gli otto passi dell'algoritmo α, di seguito elencati con i rispettivi esiti.

1. $T_L = \{a,b,c,d,e,f\}$
2. $T_I = \{a\}$
3. $T_O = \{f\}$
4. $X_L = \{(\{a\},\{b\}),(\{a\},\{e\}),(\{b\},\{c\}),(\{b\},\{f\}),(\{c\},\{d\}),(\{d\},\{b\}),(\{e\},\{f\}),(\{a,d\},\{b\}),$ $(\{b\},\{c,f\})\}$
5. $Y_L = \{(\{a\},\{e\}),(\{c\},\{d\}),(\{e\},\{f\}),(\{a,d\},\{b\}),(\{b\},\{c,f\})\}$
6. $P_L = \{p_{(\{a\},\{e\})}, p_{(\{c\},\{d\})}, p_{(\{e\},\{f\})}, p_{(\{a,d\},\{b\})}, p_{(\{b\},\{c,f\})}, i_L, o_L\}$

[36] W.M.P. van der Aalst, op. cit., p. 135.

7. $F_L = \{(a, p_{(\{a\},\{e\})}), (p_{(\{a\},\{e\})}, e), (c, p_{(\{c\},\{d\})}), (p_{(\{c\},\{d\})}, d), (e, p_{(\{e\},\{f\})}), (p_{(\{e\},\{f\})}, f),$
$(a, p_{(\{a,d\},\{b\})}), (d, p_{(\{a,d\},\{b\})}), (p_{(\{a,d\},\{b\})}, b), (b, p_{(\{b\},\{c,f\})}), (p_{(\{b\},\{c,f\})}, c),$
$(p_{(\{b\},\{c,f\})}, f), (i_L, a), (f, o_L)\}$
8. $\propto (L) = (P_L, T_L, F_L)$

La Figura 1.4 mette in evidenza la rete $N_5 = \propto (L_5)$ ricavata dall'algoritmo; N_5 può infatti replicare le tracce registrate in L_5.

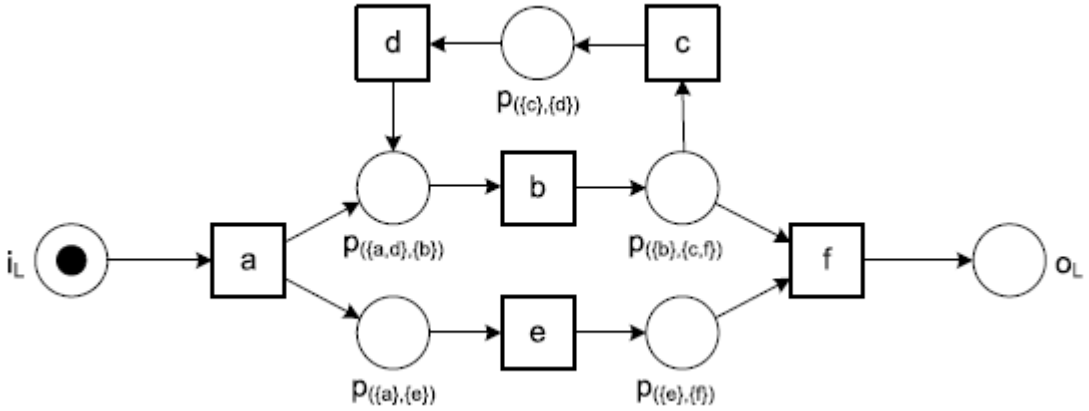

Figura 1.4 La WF-net N_5 derivata dall'applicazione dell'algoritmo α con il log L_5.

Tuttavia, l'algoritmo α presenta dei problemi. Infatti, ci sono molte reti workflow, fra loro diverse, che possono manifestare il medesimo comportamento, ossia nonostante abbiano strutture differenti possono riprodurre tracce uguali. Si consideri per esempio il seguente log di eventi:

$$L_6 = [(a, c, e, g)^2, (a, e, c, g)^3, (b, d, f, g)^2, (b, f, d, g)^4]$$

Sebbene il modello relativo a L_6 sia in grado di generare il comportamento osservato, la WF-net (Figura 1.5) che ne risulta appare più complessa, senza che ve ne sia bisogno. Due dei posti in ingresso di g sono ridondanti, cioè possono essere cancellati senza alterare il comportamento del modello. Tali posti, indicati con p_1 e p_2, sono chiamati nodi impliciti[37] e la loro rimozione non influisce sul set delle possibili sequenze soddisfacenti la regola di scatto. Infatti, in Figura 1.5 è mostrata solamente una delle tanti WF-nets equivalenti in termini di tracce possibili.

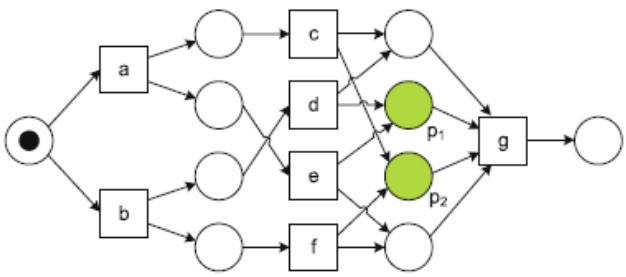

Figura 1.5 La WF-net N_6 derivata dal log di eventi L_6.

[37] W.M.P. van der Aalst, op. cit., p. 136.

Il principale problema presentato dall'algoritmo α riguarda la gestione dei cicli corti, cioè cicli di lunghezza uno o due (composti da uno o due elementi). Per quanto concerne i cicli di lunghezza uno[38], si prende come esempio il seguente log di eventi:

$$L_7 = [(a, c)^2, (a, b, c)^3, (a, b, b, c)^2, (a, b, b, b, b, c)$$

In Figura 1.6a è riportato il risultato dell'algoritmo α applicato al log L_7. Il modello, come si può osservare, non è una WF-net, in quanto la transizione b è disconnessa dal resto della rete. Inoltre, il modello consente l'esecuzione di b prima dell'attività a e dopo l'attività c, dimostrandosi incoerente con quanto descritto nel log. Ciononostante, il problema può essere risolto tramite l'utilizzo di una versione migliorata dell'algoritmo α, non oggetto di studio nel presente elaborato, il cui risultato è rappresentato in Figura 1.6b.

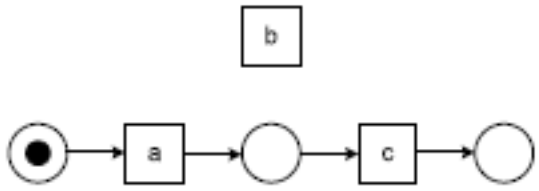

Figura 1.6a WF-net N_7 derivata da L_7 non corretta.

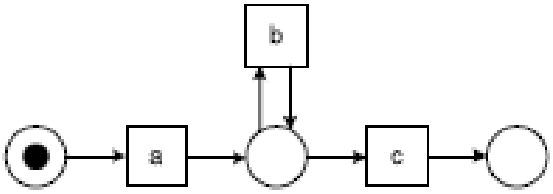

Figura 1.6b WF-net N_7' con un ciclo corto di lunghezza uno.

Per quanto riguarda i cicli di lunghezza due, vale un discorso simile; si consideri il log di eventi L_8.

$$L_8 = [(a, b, d)^3, (a, b, c, b, d)^2, (a, b, c, b, c, b, d)]$$

Le relazioni di ordinamento per L_8 sono: $a \rightarrow_{L_8} b$, $b \rightarrow_{L_8} d$ e $b \,||_{L_8} c$. Tenendo presenti tali relazioni, si può osservare da Figura 1.7 che la struttura generata dall'algoritmo α assume, in modo errato, una configurazione parallela per le attività b e c, giocando sul fatto che l'una può seguire l'altra e viceversa. Sempre in Figura 1.7, si può notare che il reticolo raffigurato non possiede le peculiarità di una WF-net; nella fattispecie, l'attività c non è compresa in nessun cammino che vada dal posto iniziale al posto finale. Adoperando la versione migliorata dell'algoritmo α, si può appianare tale criticità, ottenendo come risultato la rete workflow di Figura 1.8.

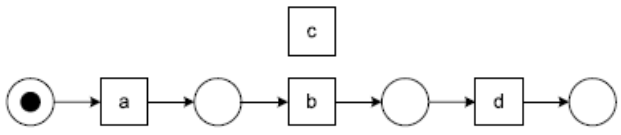

Figura 1.7 WF-net N_8 derivata da L_8 non corretta.

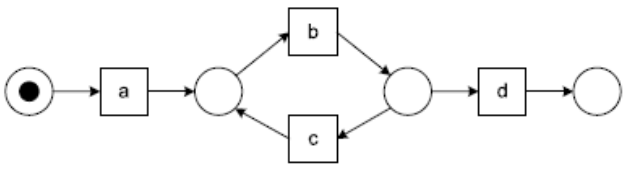

Figura 1.8 WF-net N'8 con un ciclo corto di lunghezza due.

[38] W.M.P. van der Aalst, op. cit., p. 137.

Il modello base dell'algoritmo α non mostra problemi nella gestione di cicli di lunghezza tre o superiore. Al contrario, manifesta difficoltà nella scoperta di dipendenze non locali, derivanti da costrutti con scelta obbligata. Si prenda ad esempio la rete workflow di Figura 1.9 e si consideri il log $L_9 = [(a, c, d)^{45}, (b, c, e)^{42}]$. In ogni caso, l'algoritmo α genera una WF-net che non contiene i due posti p_1 e p_2. Tuttavia, la rete di Figura 1.9, nonostante L_9 non comprenda le tracce (a, c, e) e (b, c, d), presenta quattro posti. Questi problemi, come in precedenza, possono essere risolti, anche se solo parzialmente, dalla versione migliorata dell'algoritmo α.

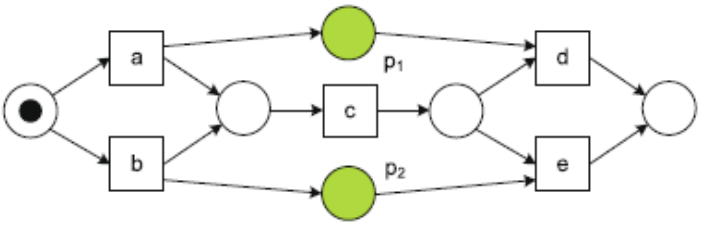

Figura 1.9 WF-net avente una dipendenza non locale.

Un altro limite dell'algoritmo α riguarda la non presa in considerazione delle frequenze; di conseguenza, è molto sensibile a incompletezza e disturbi. Questi due fenomeni[39], di seguito definiti, possono inficiare la rappresentatività di un log di eventi per finalità di studio del processo in esame:

- Disturbo: il log di eventi contiene comportamenti di rara e poco frequente occorrenza, i quali non sono rappresentativi del reale andamento del processo.
- Incompletezza: il log di eventi contiene un numero esiguo di eventi tale da non permettere la scoperta delle strutture del flusso di controllo.

L'algoritmo α presenta pertanto diversi problemi, fra i quali l'incapacità di ricavare un modello che presenti il giusto trade-off fra underfitting e overfitting. Fortunatamente, sono disponibili altri approcci al process mining; prima di affrontarli, è utile e necessario descrivere le tipiche caratteristiche degli algoritmi di process discovery[40]. La prima è il bias di rappresentazione, ossia la classe dei modelli di processo sulla quale investigare, fare deduzioni e orientare il process discovery; la funzione di tale caratteristica consiste nel determinare lo spazio di ricerca e, possibilmente, i limiti dell'espressività del modello di processo. A differenza dell'algoritmo α, sono anche ammesse notazioni diverse dalle reti di Petri (es. BPMN). Gli algoritmi, tuttavia, mostrano alcuni limiti di rappresentazione, di seguito descritti:

- Incapacità di rappresentare attività parallele
- Incapacità di gestione dei cicli (es. algoritmo α)
- Incapacità di rappresentare azioni implicite o tacite (es. saltare un'attività)

[39] W.M.P. van der Aalst, op. cit., p. 147.
[40] W.M.P. van der Aalst, op. cit., pp. 159-162.

- Incapacità di rappresentare azioni duplicate
- Incapacità di modellare congiunzioni o disgiunzioni di tipo OR
- Incapacità di rappresentare comportamenti in cui la scelta non è libera
- Incapacità di rappresentare la gerarchia

La seconda caratteristica concerne la gestione di disturbi e perturbazioni al modello. L'attenzione è rivolta ai comportamenti aventi maggiore frequenza e, pertanto, ritenuti più rappresentativi; conseguentemente, gli algoritmi fronteggiano tale problematica astraendosi dai comportamenti eccezionali e sporadici. Entrambe le prime due caratteristiche sono di notevole importanza per il buon esito di un algoritmo di process discovery. La terza caratteristica si riferisce alle assunzioni di completezza; ad esempio, l'algoritmo α assume che la relazione $>_L$ sia completa, cioè se un'attività è seguita direttamente da un'altra attività, tale situazione sarà osservabile almeno una volta nel log. Altri algoritmi adottano altre assunzioni di completezza, ad esempio considerando come completi i log di eventi che contengono al loro interno tutte le possibili tracce; in questo caso, l'assunzione di completezza è molto stringente. Assunzioni di questo tipo tendono a generare modelli affetti da overfitting, mentre assunzioni di completezza più morbide e meno vincolanti sfociano in modelli in cui si registra underfitting. L'ultima delle quattro caratteristiche riguarda l'approccio utilizzato per la conduzione del process discovery; i vari approcci sono divisi per famiglie. L'approccio diretto rappresenta la prima di queste famiglie: sostanzialmente, si estraggono alcuni footprint da un log di eventi per poi utilizzarli nella costruzione del modello di processo, alla quale contribuiscono in maniera diretta. L'algoritmo α è un esempio di tale approccio. La seconda famiglia racchiude gli approcci a due passi, dove, in principio, si costruisce un modello di base e, successivamente, lo si converte in un modello di livello più elevato. Un classico esempio è un log di eventi dal quale si estrae un sistema di transizione (modello base) che poi sarà convertito in una rete di Petri (modello di livello più alto). La terza famiglia è relativa agli approcci legati alla computational intelligence, dalla quale si originano diverse tecniche con un aspetto comune: il log non è convertito direttamente in un modello ma segue una procedura iterativa per imitare il processo di valutazione naturale. Un esempio di questo approccio è il genetic process mining. Ognuno dei tre approcci è descritto più in dettaglio di seguito nel presente elaborato; a ciascuno di essi è associata un tecnica di process mining. Si annovera nella famiglia degli approcci diretti, oltre all'algoritmo α, un'altra tecnica nota come heuristic mining[41], la quale usa una rappresentazione simile alle reti casuali (C-nets). In Figura 1.10 è mostrata una rete casuale, nella quale i nodi e gli archi rappresentano rispettivamente le attività e le dipendenze causali. Ogni attività è caratterizzata da insiemi di input bindings e output bindings (legami in ingresso e in uscita); per esempio, nella Figura 1.10 l'attività a non prevede input bindings, essendo la prima attività della rete. Presenta invece tre possibili output bindings: e, $[b, c]$ e d. Vuol dire che a è seguita da b e c, oppure da

[41] W.M.P. van der Aalst, op. cit., pp. 163-167.

e o in ultima ipotesi da d. La rete di Petri e la rete casuale sono equivalenti in termini di tracce, ossia consentono la rappresentazione dello stesso insieme di cammini. Tuttavia, la rete casuale non contiene posti; la sua logica di percorrenza è determinata solamente da input e output bindings. Le sequenze, nell'esempio in questione, sono considerate valide solamente se iniziano dall'attività $a_i = a$ e terminano con l'attività $a_i = e$; inoltre, devono rimuovere gli obblighi pendenti, giungendo all'attività finale con nessuno di essi ancora irrisolto. Dopo aver eseguito $(a, \emptyset, \{b, c\})$, la cui notazione è (attività eseguita, input bindings, output bindings), sono presenti due obblighi pendenti: (a, b) e (a, c). Pertanto b e c vedranno l'attività a come un loro input bindings. Eseguendo l'attività b si elimina l'obbligo pendente (a, b), me si crea (b, e) come nuovo obbligo, ecc. I modelli di processo come quelli in Figura 1.10 non possono essere espressi con le WF-net; per il process discovery le C-nets sono la rappresentazione più adatta. Per illustrare i concetti basilari dell'heuristic mining, si consideri il seguente evento:

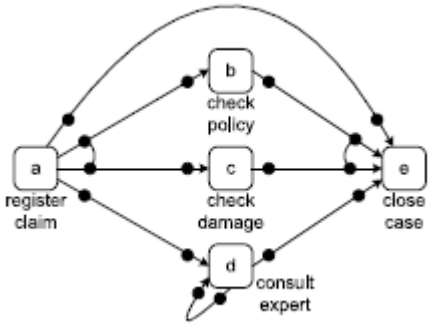

Figura 1.10 Esempio di rete casuale utilizzata per l'heuristic mining.

$$L = [(a, e)^5, (a, b, c, e)^{10}, (a, c, b, e)^{10}, (a, b, e), (a, c, e), (a, d, e)^{10}, (a, d, d, e)^2, (a, d, d, d, e)]$$

Assumendo che le 3 tracce aventi frequenza 1 siano dei disturbi, le restanti 37 tracce sono considerabili come valide sequenze della rete casuale in Figura 1.10. Applicando l'algoritmo α, si nota che il modello non permette la rappresentazione di molte delle tracce del log L, fra le quali ve ne sono anche di molto frequenti come (a, e) e (a, d, e); tali considerazioni risultano immediate osservando la Figura 1.11. I motivi per i quali l'algoritmo α esprime simili comportamenti sono da ricondurre a due problemi: il suo bias di rappresentazione non permette di saltare le attività (es. non è in grado di descrivere un salto da a ad e) e non riesce a gestire, nella fattispecie, il caso dell'attività d, che deve essere eseguita almeno una volta se presente nelle tracce. Il secondo problema è relativo alla mancanza di considerazione verso le frequenze. Al contrario, le C-nets tengono conto delle frequenze e, pertanto, sono lo strumento più adatto a fini di heuristic mining. In Tabella 1.6 sono riassunte le frequenze con le quali un'attività è seguita direttamente da un'altra attività. Con tali valori è possibile calcolare le relazioni di dipendenza intercorrenti fra ciascuna coppia di attività, secondo la seguente formula:

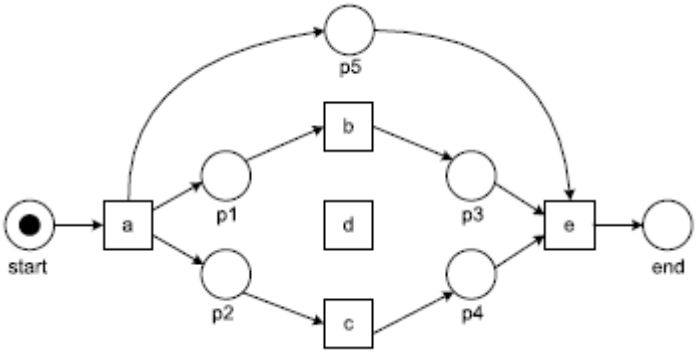

Figura 1.11 WF-net del log L realizzata con l'algoritmo α.

$$|a >_L b| = \sum L(\sigma) \times |\{1 \leq i < |\sigma| \; tale \; che \; \sigma(i) = a \wedge \sigma(i+1) = b\}|$$

La relazione $|a >_L b|$ indica il numero di volte in cui l'attività a è seguita direttamente in sequenza dall'attività b. La relazione $|a \Rightarrow_L b|$, invece, esprime il valore della relazione di dipendenza[42] esistente fra le attività a e b, secondo la seguente funzione a tratti:

$$|a \Rightarrow_L b| = \begin{cases} \dfrac{|a >_L b| - |b >_L a|}{|a >_L b| + |b >_L a| + 1} & se\ a \neq b \\ \dfrac{|a >_L a|}{|a >_L a| + 1} & se\ a = b \end{cases}$$

$|a \Rightarrow_L b|$ assume valori compresi fra 1 e -1: nel caso in cui abbia valore vicino a 1, intercorre una forte dipendenza positiva fra a e b, ossia a è spesso la causa di b. Per valori prossimi a -1, intercorre una forte dipendenza negativa fra a e b, cioè b è spesso la causa di a. Nel caso speciale $|a \Rightarrow_L a|$, ossia un'attività è seguita da sé stessa, si assume per definizione $\frac{|a>_L b|-|b>_L a|}{|a>_L b|+|b>_L a|+1} = 0$. In Tabella 1.7 sono riportate le misure di dipendenza relative al log L.

Tabella 1.6 Frequenze con cui un'attività segue direttamente un'altra attività relative al log $\boldsymbol{L}$.

$\|>_L\|$	a	b	c	d	e
a	0	11	11	13	5
b	0	0	10	0	11
c	0	10	0	0	11
d	0	0	0	4	13
e	0	0	0	0	0

Tabella 1.7 Misure di dipendenza fra le attività $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$ *del log* $\boldsymbol{L}$.

$\|\Rightarrow_L\|$	a	b	c	d	e
a	$\frac{0}{0+1} = 0$	$\frac{11-0}{11+0+1} = 0.92$	$\frac{11-0}{11+0+1} = 0.92$	$\frac{13-0}{13+0+1} = 0.93$	$\frac{5-0}{5+0+1} = 0.83$
b	$\frac{0-11}{0+11+1} = -0.92$	$\frac{0}{0+1} = 0$	$\frac{10-10}{10+10+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{11-0}{11+0+1} = 0.92$
c	$\frac{0-11}{0+11+1} = -0.92$	$\frac{10-10}{10+10+1} = 0$	$\frac{0}{0+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{11-0}{11+0+1} = 0.92$
d	$\frac{0-13}{0+13+1} = -0.93$	$\frac{0-0}{0+0+1} = 0$	$\frac{0-0}{0+0+1} = 0$	$\frac{4}{4+1} = 0.8$	$\frac{13-0}{13+0+1} = 0.93$
e	$\frac{0-5}{0+5+1} = -0.83$	$\frac{0-11}{0+11+1} = -0.92$	$\frac{0-11}{0+11+1} = -0.92$	$\frac{0-13}{0+13+1} = -0.93$	$\frac{0}{0+1} = 0$

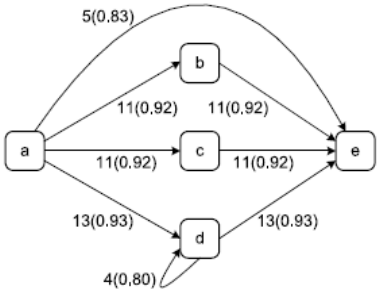

Figura 1.12 Dependency graph con soglie 2 e 0.7.

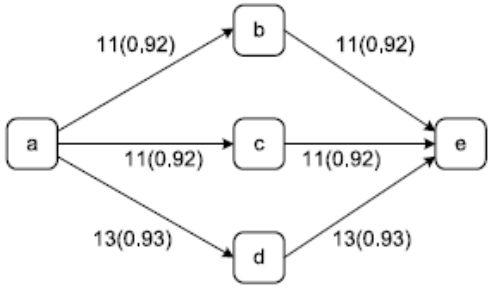

Figura 1.13 Dependency graph con soglie 5 e 0.9.

[42] A.J.M.M. Weijters, W.M.P. van der Aalst, e A.K. Alves de Medeiros, Process Mining with the HeuristicsMiner Algorithm, Department of Technology Management, Eindhoven University of Technology, 2006, pp. 7-9.

Dalle informazioni presenti nelle Tabelle 1.6 e 1.7 si può derivare il dependency graph (o grafo delle dipendenze); in Figura 1.12 ne è mostrato un esempio riferito al log L con valori di soglia 2 e 0.7 rispettivamente per $|>_L|$ e $|\Rightarrow_L|$. Ciò significa che si considerano solo gli archi congiungenti due attività x e y se $|x >_L y| \geq 2$ e $|x \Rightarrow_L y| \geq 0.7$. Utilizzando invece valori di soglia maggiori, ad esempio 5 e 0.9, si ottiene un dependency graph più stringente (Figura 1.13) e che, nel caso specifico, perde due archi rispetto alla configurazione precedente. Il dependency graph non mostra quale sia la logica di percorrenza, cioè quali attività y possono seguire una certa attività x; tuttavia, è in grado di evidenziare quale sia la struttura portante del modello di processo. Modificando i valori di soglia, è possibile rendersi conto di quali siano i comportamenti meno frequenti e se catalogarli come disturbi. Per escludere tali comportamenti è però necessario riprocessare il log di eventi, eliminando le attività più sporadiche; solo dopo si può ricreare il nuovo grafo delle dipendenze. Dopo aver descritto modelli di processi che utilizzano approcci diretti e deterministici, quali l'algoritmo α e l'heuristic mining, si procede illustrando un esempio di approccio al process discovery che prende spunto dalle tecniche provenienti dal campo della computational intelligence: il genetic process mining[43]. Qualsiasi algoritmo di tipo genetico si sviluppa in quattro fasi: inizializzazione, selezione, riproduzione e conclusione[44]. La Figura 1.14 mostra una visione d'insieme dell'approccio utilizzato per il genetic process mining. Nella fase di inizializzazione si crea la popolazione, che sarà la prima generazione di individui ad essere usata; nel caso specifico, gli individui sono i modelli di processo. Servendosi dei nomi delle attività presenti nel log, si generano modelli di processo in modo casuale. Ogni generazione potrebbe contare centinaia o addirittura migliaia di modelli, i quali probabilmente presenteranno comportamenti discordanti da quelli osservati nel log; tuttavia, possono esserci modelli che, in parte, ben si adattano al log degli eventi in esame, per ragioni da ricondurre alla casualità e alla grande numerosità degli individui. Nella selezione si calcola la bontà di ogni individuo; tramite una funzione di fitness si determina la qualità dell'individuo rispetto al log di eventi. Tale funzione non deve

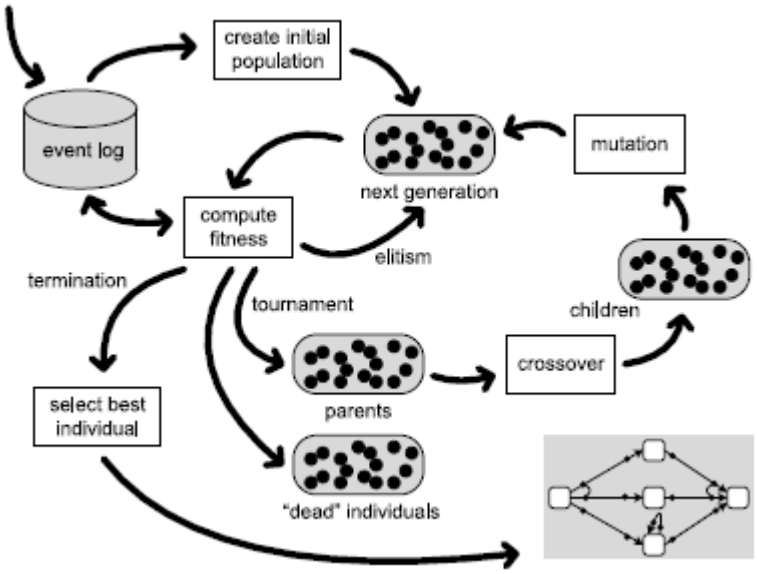

Figura 1.14 Panoramica dell'approccio usato per il genetic process mining.

[43] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, pp. 169-173.

[44] W.M.P. van der Aalst, A.K. Alves de Medeiros, e A.J.M.M. Weijters, Genetic Process Mining, Department of Technology Management, Eindhoven University of Technology, 2006, pp. 61-65.

essere eccessivamente generalista, deve premiare la parziale correttezza di un modello e tenere in considerazione i quattro criteri qualitativi in precedenza descritti. Gli individui con i più alti valori di fitness relativamente al log, detti best individuals, passano alla generazione successiva senza subire alcuna modifica; quest'ultima procedura prende il nome di elitismo. Attraverso tornei, si selezionano gli individui "genitori" con i quali crearne degli altri. L'elitismo e i tornei fra gli individui garantiscono con alta probabilità che il materiale genetico dei modelli di processo migliori sia impiegato per la successiva generazione. Ne consegue che gli individui con un fitness non soddisfacente non sopravvivranno alle generazioni successive; si fa riferimento a tali modelli con il termine individui morti. Per quanto riguarda la fase di riproduzione, gli individui genitori selezionati sono adoperati per la creazione di nuove progenie. A tale fine si utilizzano due operatori genetici: il crossover e la mutazione. Il primo consiste nel generare due modelli "figli" a partire da due modelli genitori, dei quali posseggono parte del materiale genetico. Tali individui figli sono poi modificati tramite la mutazione, la quale consta dell'aggiunta o dell'eliminazione in modo casuale di una dipendenza causale. Con la mutazione si inserisce nuovo materiale genetico nella generazione successiva; senza tale pratica, l'evoluzione oltre il materiale genetico della popolazione iniziale sarebbe impossibile. In definitiva, attraverso la riproduzione (costituita da crossover e mutazione) e l'elitismo, si crea una nuova generazione. Da qui, si itera il procedimento sopra descritto, calcolando il fitness, promuovendo i best individuals, ecc. Il processo di evoluzione termina quando si trova un modello in possesso del fitness desiderato (ultima fase). In base al log analizzato, potrebbe essere necessario molto tempo affinché il modello converga al livello di fitness desiderato, senza mai raggiungerlo in alcuni casi. Per tale motivo, occorre definire alcuni criteri di conclusione, come ad esempio un numero massimo di generazioni o un numero di generazioni consecutive nelle quali non si producono individui migliori dei precedenti. L'approccio sopra descritto e raffigurato in Figura 1.14 è molto generale. Solitamente, in fase di implementazione di un algoritmo relativo al genetic process mining si effettuano scelte ben precise, di seguito elencate:

- Rappresentazione degli individui. Ogni individuo, come già anticipato, corrisponde a un modello di processo descritto con una particolare notazione (es. reti di Petri, C-nets, BPMN, ecc.). Tale scelta è importante in quanto determina la classe dei processi che possono essere scoperti, ossia determina il bias di rappresentazione.
- Inizializzazione. La casualizzazione dei modelli di processo per generare la popolazione iniziale può avvenire secondo due approcci: nel primo si inserisce una certa probabilità di dipendenza causale fra due attività in modo da creare le C-nets. L'altro approccio prevede l'uso di una variante casualizzata dell'heuristic mining per la creazione di una popolazione avente un livello medio di fitness superiore rispetto a quanto si ottiene con il primo approccio.

- Funzione di fitness. L'obiettivo è la definizione di una funzione equilibrata rispetto ai quattro criteri qualitativi. La funzione di fitness guida il processo di evoluzione e può essere a favore di specifiche tipologie di modello.
- Strategia di selezione. La selezione degli individui genitori può avvenire secondo vari approcci. Ad esempio, si possono adottare tornei con cinque individui ciascuno; il vincitore di ogni torneo sarà adoperato come individuo genitore. Per cui si selezionano casualmente cinque modelli di processo e si sceglie come genitore quello che presenta un livello di fitness maggiore.
- Crossover. Si prendono due individui genitori con lo stesso insieme di attività, delle quali se ne sceglie una. Si procede quindi con lo scambio parziale degli input e output bindings. Così facendo si possono generare due individui figli.
- Mutazione. Si seleziona un'attività di ogni individuo figlio e si aggiungono o si cancellano da essa bindings potenziali, con modalità casuali. L'obiettivo è l'inserimento casuale di materiale genetico nuovo.

Il genetic process mining è sia robusto sia flessibile. In modo simile alle tecniche di heuristic mining, è in grado di gestire i disturbi e l'incompletezza. Variando la funzione di fitness è possibile dare preferenza a particolari modelli e costrutti. Sfortunatamente, il genetic process mining non è molto efficiente in presenza di modelli e log di grandi dimensioni. Richiede molto tempo per arrivare alla scoperta di un modello con un adeguato livello di fitness; tuttavia, può essere combinato con l'heuristic mining. In particolare, il genetic process mining è utilizzato per migliorare il modello di processo ottenuto in seguito all'applicazione dell'heuristic mining.

L'analisi si conclude con l'approccio a due passi, sul quale si fonda il region-based mining[45]. Tale tecnica può essere applicata secondo due diverse metodologie: lo state-based regions e il language-based regions. Lo state-based regions permette la costruzione di una rete di Petri, derivandola da un sistema di transizione, mentre il language-based regions ottiene una rete di Petri partendo però da particolari linguaggi noti come prefix-closed. Nel presente elaborato, per ragioni di attinenza a quanto descritto fin qui, sarà illustrata solamente la prima tipologia, cioè lo state-based regions[46]. Prima di tutto, occorre definire un sistema di transizione, il quale si basa sulla tracce registrate all'interno di un log di eventi. Specificatamente, un sistema di transizione è descritto da una tripla $TS = (S, A, T)$, in cui S è l'insieme degli stati, A è l'insieme delle attività e $T \subseteq S \times A \times S$ è l'insieme delle transizioni. Inoltre, si denotano con $S^{start} \subseteq S$ l'insieme degli stati iniziali e con $S^{end} \subseteq S$ l'insieme degli stati finali. Si consideri ora $\sigma' = (a, b, c, d, c, d, c, d, e, f, a, g, h, h, h, i) \in L$ come una generica traccia del log L. Ogni posizione della traccia, quindi prima dell'evento iniziale, in mezzo a due eventi e dopo l'ultimo evento,

[45] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, pp. 173-180.

[46] W. M. P. van der Aalst, V. Rubin, H. M.W. Verbeek, B. F. van Dongen, E. Kindler e C. W. Günther, Process mining: a two-step approach to balance between underfitting and overfitting, Springer-Verlag, Berlino, 2008, pp. 94-102.

corrisponde ad uno stato del sistema di transizione. Considerando l'esempio di Figura 1.15, si possono distinguere due tracce parziali: la traccia $\sigma'_{past} = (a, b, c, d, c, d, c, d, e)$ descrive il passato del caso in esame, mentre la traccia $\sigma'_{future} = (f, a, g, h, h, h, i)$ ne descrive il futuro. La funzione di rappresentazione degli stati $l^{state}()$, dati una sequenza σ e un valore k indicante il numero di eventi occorsi relativi alla traccia σ, produce stati corrispondenti alle attività occorse nei primi k eventi. Si consideri una generica traccia $\sigma = (a_1, a_2, \dots, a_n) \in L$ di lunghezza n; $l_1^{state}(\sigma, k) = hd^k(\sigma) = (a_1, a_2 \dots, a_k)$ è un esempio di funzione di rappresentazione degli stati. La funziona restituisce la "testa" (head) della sequenza σ, ossia i primi k elementi; in pratica, descrive lo stato attuale e la "storia" del caso in esame dopo k eventi. Per la traccia σ', ad esempio, si ottiene $l_1^{state}(\sigma', 9) = (a, b, c, d, c, d, c, d, e)$. Si consideri il seguente log di eventi: $L_1 = [(a, b, c, d)^3, (a, c, b, d)^2, (a, e, d)]$. Prendendo la traccia $\sigma = (a, b, c, d)$, si ha come stato iniziale $l_1^{state}(\sigma, 0) =()$. Dopo l'esecuzione dell'attività a lo stato evolve in $l_1^{state}(\sigma, 1) = (a)$; in seguito all'esecuzione dell'ultimo evento d, lo stato risulta essere $l_1^{state}(\sigma, 4) = (a, b, c, d)$. I cinque stati visitati e le corrispondenti transizioni si aggiungono sul sistema di transizione; la stessa operazione è ripetuta per le tracce (a, c, b, d) e (a, e, d). Il risultato finale consiste nel sistema di transizione $TS_{L_1, l_1^{state}}$ rappresentato in Figura 1.16. Un sistema di transizione definisce un modello di processo di livello base, ossia rappresenta il primo dei due passi di tale approccio. Fortunatamente, lo state-based regions consente di sintetizzare un sistema di transizione in un modello più compatto e di livello più alto, nella fattispecie in una rete di Petri. L'idea di fondo consiste nella scoperta delle regioni**,** corrispondenti ai posti della rete di Petri. La regione $R \subseteq S$ è un insieme di stati. In particolare, R è un regione se per ogni attività $a \in A$ sussiste una delle seguenti condizioni:

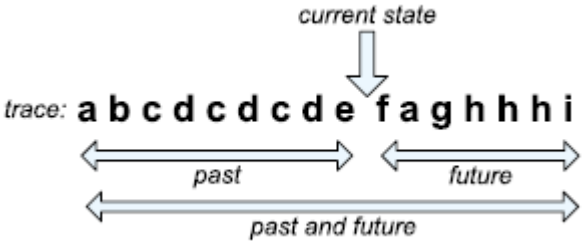

Figura 1.15 La traccia σ' con i suoi stati passati e futuri.

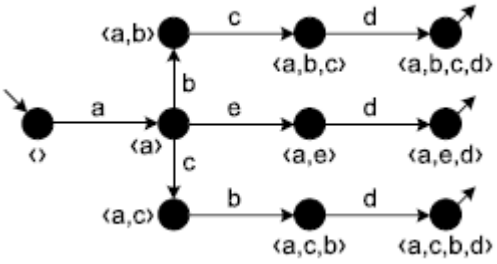

Figura 1.16 Sistema di transizione $TS_{L_1, l_1^{state}}$ derivato da L_1.

- Tutte le transizioni $(s_1, a, s_2) \in T$ entrano in R, ossia $s_1 \notin R$ e $s_2 \in R$.
- Tutte le transizioni $(s_1, a, s_2) \in T$ escono da R, ossia $s_1 \in R$ e $s_2 \notin R$.
- Tutte le transizioni $(s_1, a, s_2) \in T$ non attraversano R, ossia $s_1, s_2 \in R$ o $s_1, s_2 \notin R$.

Un'attività che entra nella regione R in una parte del sistema di transizione non può uscire da un'altra parte del sistema stesso; l'esempio di Figura 1.17 aiuta a comprendere meglio questo aspetto. Il rettangolo tratteggiato definisce la regione R; il posizionamento di tutte le attività è in riferimento alla regione medesima. Come si può osservare, le transizioni a e b sono entranti in R, le transizioni c e d

sono uscenti da C e le transizioni e e f non attraversano R. Per definizione, l'unione di due regioni è anch'essa una regione; per tale motivo si considerano solo le regioni minimali. Come già accennato precedentemente, ogni regione minimale corrisponde ad un posto p_R nella rete di Petri (Figura 1.17), Le attività entranti in R diventano transizioni di una rete di Petri avente p_R come posto in uscita, le attività uscenti da R diventano transizioni di una rete di Petri in uscita dal posto p_R e le attività che non attraversano R diventano transizioni di una rete di Petri non connesse al posto p_R. Perciò si può affermare che le regioni codificano interamente una rete di Petri. In Figura 1.18 è mostrato un esempio di approccio a due passi, dove prima si costruisce un sistema di transizione e, successivamente, si ricava da esso la rete di Petri corrispondente. Tale esempio, applicato al log di eventi L_1, considera sei regioni minimali; di conseguenza, saranno sei anche i posti della rete di Petri. Si consideri la regione minimale $R_1 = \{[a], [a, c]\}$. Tutte le transizioni a sono entranti in R_1 (una sola transizione), tutte le transizioni b sono uscenti da R_1 (due transizioni), tutte le transizioni e sono uscenti da R_1 (una sola transizione); le restanti transizioni non attraversano R_1. Ne consegue che la regione R_1 sia corrispondente al posto p_1, con in ingresso la transizione a e in uscita le transizioni b e e. Lo stesso procedimento è ripetuto per le altre cinque regioni minimali. Come risultato si ottiene la rete di Petri esibita in Figura 1.18, avente sei posti come anticipato. Tale approccio a due passi è applicabile anche per processi di notevole grandezza con diverse attività eseguite in contemporanea; tali processi possono essere rappresentati con dimensioni molto ridotte. Basti pensare che un sistema di transizione rappresentante 10 attività parallele è costituito da $2^{10} = 1024$ stati e da $10 \times 2^{10-1} = 5120$ transizioni; convertendolo in una rete di Petri, i valori si riducono a 20 posti e 10 transizioni. Tuttavia, per i grandi processi, la funzione di rappresentazione degli stati nella forma $l_1^{state}()$ restituisce un modello affetto da overfitting che può replicare solamente il log in esame, senza alcuna possibilità di generalizzazione. Ciononostante, sono molte le astrazioni possibili con le quali creare equilibrio fra le condizioni di underfitting e di overfitting.

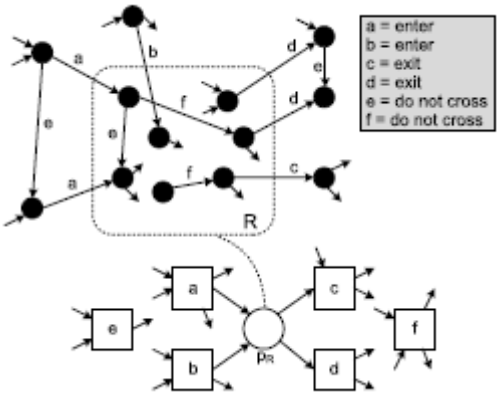

Figura 1.17 La regione R corrispondente al posto p_R.

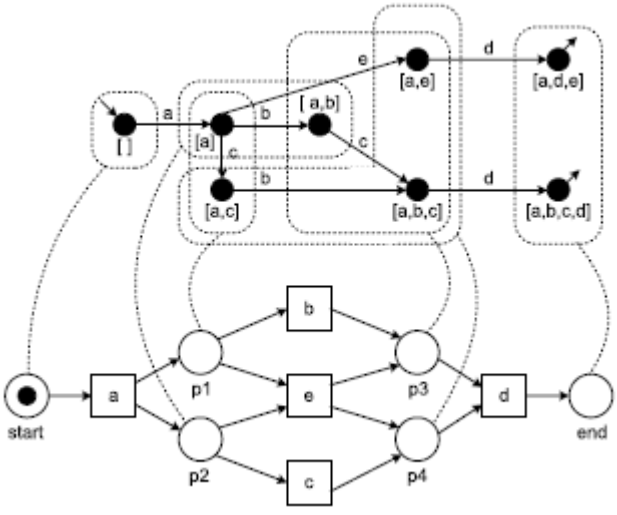

Figura 1.18 Sistema di transizione derivato dal log L_1 e convertito in una rete di Petri utilizzando la tecnica dello state-based regions.

2. Conformance checking

Sinora sono state illustrate le varie tecniche di process mining utili alla costruzione dei modelli. L'attenzione ora si rivolge al controllo della qualità di tali modelli; a tale scopo è destinata la fase di conformance checking[47], la quale mette in relazione gli eventi contenuti nel log con le attività del modello realizzato, al fine di compararli. L'obiettivo consiste nell'individuare le analogie e le discrepanze fra il comportamento modellato e il comportamento osservato. Il conformance checking è di grande rilevanza per l'allineamento del business e per le operazioni di revisione e verifica ad esso collegate. Ad esempio, le tecniche di conformance checking sono fruibili per la misurazione delle performance riguardanti gli algoritmi ottenuti a seguito del process discovery; così facendo, è possibile rettificare i modelli che non si dimostrino allineati con la realtà. In precedenza, erano stati definiti i termini play-in, play-out e replay; la terza soluzione, il replay appunto, è quella utilizzata dal conformance checking, di cui è mostrata l'idea di base in Figura 2.1. I risultati delle analisi di confronto fra i comportamenti manifestati dal modello e quelli osservati nel log sono di due tipologie: le misure di conformità globale (global conformance measures) e le diagnosi a livello locale (local diagnostics). Gli esiti del primo tipo riguardano il processo nella sua totalità (es. l'85% dei casi presenti nel log di eventi in esame possono essere riprodotti dal modello), mentre le diagnosi locali sono più specifiche e puntuali (es. l'attività x è stata eseguita 15 volte sebbene il modello, per come è configurato, non lo consenta). L'interpretazione delle non conformità dipende dallo scopo al quale il modello è destinato; nel caso in cui il modello sia di tipo descrittivo, le discrepanze sono indicative di una necessità di miglioramento del modello stesso, in modo da cogliere più adeguatamente la realtà. In presenza, invece, di un modello normativo, le discrepanze possono avere una doppia interpretazione. Infatti, in tal caso si può distinguere fra deviazioni indesiderate (es. il conformance checking segnala l'esigenza di controllare il processo in maniera migliore) e deviazioni desiderate (es. i lavoratori sono in grado di servire i clienti e gestire le circostanze in modo più tempestivo e appropriato di quanto previsto dal modello di processo). Non a caso, le non conformità e la flessibilità del processo presentano spesso una correlazione positiva. È

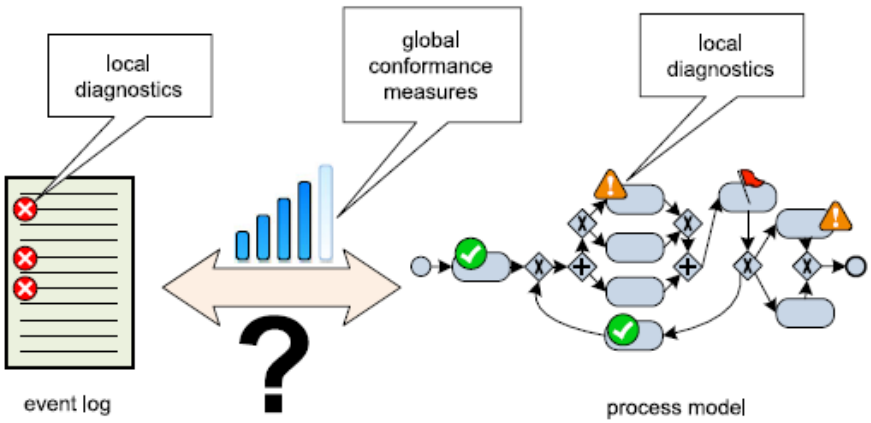

Figura 2.1 Il processo di comparazione fra i comportamenti osservati e quelli modellati, il quale costituisce il fondamento del conformance checking.

[47] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, pp. 191-213.

importante sottolineare che, in fase di conformance checking, le deviazioni rilevate devono essere osservate secondo due diversi punti di vista, cercando di dare una risposta alle seguenti domande:

1. Il modello è errato e non riflette fedelmente la realtà. Come lo si può migliorare?
2. I casi osservati deviano dal modello di processo creato e, pertanto, si rende indispensabile apportare dei correttivi. Come si può migliorare il controllo in modo da ottenere un livello di conformità più elevato?

Le tecniche di conformance checking dovrebbero essere abili a supportare entrambe le prospettive; per tale ragione, la Figura 2.1 mostra alcune discrepanze su ambo i lati. L'allineamento del business si propone di ottenere sistemi informativi capaci di riflettere correttamente i processi reali; tuttavia, spesso si riscontrano discrepanze che sfavoriscono il raggiungimento di tale proposito. Le cause sono da ricondurre a diversi motivi; innanzitutto, gran parte delle aziende utilizza software generici, non sviluppati per una singola organizzazione. Un classico esempio è rappresentato dal sistema SAP, il quale si basa sulle cosiddette "best practices", ossia si implementano nel sistema i processi e gli scenari tipici. Sebbene tali sistemi siano configurabili secondo le proprie richieste, le particolari esigenze di una certa organizzazione possono essere profondamente differenti rispetto a quanto immaginato dallo sviluppatore del software. In secondo luogo, i processi variano più rapidamente dei sistemi informativi, principalmente a causa dell'influenza dell'ambiente esterno; è pertanto necessario che il sistema informativo mantenga lo stesso passo del processo in esame al fine di un buon allineamento fra di essi. Infine, gli stakeholder di un'organizzazione possono presentare interessi in conflitto l'uno con l'altro, aspetto che non aiuta la concretizzazione di un buon allineamento. Oltre alla ricerca di un allineamento adeguato, è importante eseguire operazioni di revisione e di valutazione all'interno dell'organizzazione, in modo da accertare la validità e l'affidabilità delle informazioni registrate a sistema, specie quelle associate ai processi. Oggigiorno, le informazioni relative ai processi sono conservate in database, data warehouse, log di eventi, ecc. Per tale ragione, non è più sufficiente né accettabile controllare solamente un campione ristretto di informazioni, come accadeva in passato. Con il supporto dei sistemi informativi, ora è possibile valutare tutti gli eventi di un processo operativo, persino mentre il processo è in stato di esecuzione. Il process mining in generale, ma più in particolare il conformance checking, è in grado di fornire i mezzi per effettuare questo tipo di azioni. In riferimento ai quattro criteri qualitativi, la nozione di fitness è strettamente legata al conformance checking e può essere quantificata. Il fitness misura in che proporzione il comportamento osservato nel log sia correttamente rappresentato dal modello. Per spiegare meglio il significato, si consideri il log di eventi L_{full}, composto dalle attività a, b, c, d, e, f, g, h. Il log è riferito all'esempio riguardante la richiesta di risarcimento, già adoperato in precedenza. Dalla Tabella 2.1 si nota la presenza di 1391 casi all'interno di L_{full}, distribuiti lungo 21 tracce diverse. Ad esempio, vi sono 455 casi che seguono la traccia $\sigma_1 = (a, d, c, e, h)$, 191 casi che seguono la traccia $\sigma_2 = (a, b, d, e, g)$, 177 casi che seguono la traccia $\sigma_3 = (a, d, c, e, h)$, ecc.

Tabella 2.1 Log di eventi L_{full} relativo alla richiesta di risarcimento.

Frequenza	Referenza	Traccia
455	σ_1	(a, c, d, e, h)
191	σ_2	(a, b, d, e, g)
177	σ_3	(a, d, c, e, h)
144	σ_4	(a, b, d, e, h)
111	σ_5	(a, c, d, e, g)
82	σ_6	(a, d, c, e, g)
56	σ_7	(a, d, b, e, h)
47	σ_8	$(a, c, d, e, f, d, b, e, h)$
38	σ_9	(a, d, b, e, g)
33	σ_{10}	$(a, c, d, e, f, b, d, e, h)$
14	σ_{11}	$(a, c, d, e, f, b, d, e, g)$
11	σ_{12}	$(a, c, d, e, f, d, b, e, g)$
9	σ_{13}	$(a, d, c, e, f, c, d, e, h)$
8	σ_{14}	$(a, d, c, e, f, d, b, e, h)$
5	σ_{15}	$(a, d, c, e, f, b, d, e, g)$
3	σ_{16}	$(a, c, d, e, f, b, d, e, f, d, b, e, g)$
2	σ_{17}	$(a, d, c, e, f, d, b, e, g)$
2	σ_{18}	$(a, d, c, e, f, b, d, e, f, b, d, e, g)$
1	σ_{19}	$(a, d, c, e, f, d, b, e, f, b, d, e, h)$
1	σ_{20}	$(a, d, b, e, f, b, d, e, f, d, b, e, g)$
1	σ_{21}	$(a, d, c, e, f, d, b, e, f, c, d, e, f, d, b, e, g)$

In Figura 2.2 sono mostrate quattro reti workflow relative al log L_{full}. La WF-net N_1è un modello di processo derivato in seguito all'applicazione dell'algoritmo α. LA WF-net N_2 è un modello sequenziale dove, rispetto a N_1, l'attività d è sempre preceduta da b o da c; N_2 non consente la riproduzione di tutte le tracce presenti nel log. Ad esempio, la traccia $\sigma_3 = (a, d, c, e, h)$ non è rappresentabile dalla rete N_2. La rete N_3 presenta una configurazione che non lascia possibilità di scelta, nel senso che la richiesta sarà, in qualunque caso, rigettata. Chiaramente, anche la WF-net N_3 non permette la descrizione di alcune delle tracce del log L_{full}, come per esempio accade per $\sigma_2 = (a, b, d, e, g)$. La rete workflow N_4 presenta una conformazione tale da rappresentare tutte le tracce presenti in L_{full}. Un semplice approccio verso il conformance checking consiste nel calcolare il fitness come rapporto fra tracce rappresentabili e tracce totali. Così facendo, la rete N_1 presenta fitness pari a $\frac{1391}{1391} = 1$, ossia tutte le tracce sono riproducibili da tale rete, come prima anticipato. I valori di fitness che si ottengo per le WF-nets N_2, N_3 e N_4 sono pari rispettivamente a $\frac{948}{1391} = 0.6815$, $\frac{632}{1391} = 0.4543$ e $\frac{1391}{1391} = 1$.

Occorre sottolineare che una metrica di fitness così poco complessa non è adatta per i processi reali. Si consideri ad esempio una variante della rete workflow N_1 avente i due posti p_1 e p_2 accorpati in un unico posto. Un simile modello avrebbe un valore di fitness corrispondente a $\frac{0}{1391} = 0$, perché nessuna delle tracce può essere replicata. Ciò significa che, utilizzando l'approccio semplicista sopra descritto, potrebbe accadere di dichiarare un certo modello come "non in fitting" a causa di un solo evento non riproducibile (es. 99 eventi su 100 sono riproducibili; 10 eventi su 100 sono riproducibili; tale approccio porta a concludere che entrambi i modelli siano da etichettare come "non in fitting" con il log di eventi in esame). In seguito a quanto detto, è pertanto necessario utilizzare una nozione di fitness definita a livello di eventi invece che di tracce complete. Con l'approccio precedente, si interrompeva la replicazione di una traccia alla prima manifestazione di un problema e la si giudicava come "non in fitting". Ora, invece, si continua la riproduzione della traccia sul modello senza fermarla, ma registrando tutte le situazioni in cui una transizione è forzata al rispetto della regola di scatto pur non essendo abilitata; in sintesi, si opera un conteggio dei token mancanti. Inoltre, si contano anche i token rimanenti a fine replicazione. Per chiarire in modo più approfondito questo diverso approccio, si può considerare come esempio esplicativo la replicazione della traccia σ_1 sulla rete N_1. Come si può evincere dalla Tabella 2.1 e dalla Figura 2.2, la traccia σ_1 può essere interamente riprodotta. Nelle Figure 2.3a e 2.3b sono mostrati i vari passaggi relativi alla replicazione di

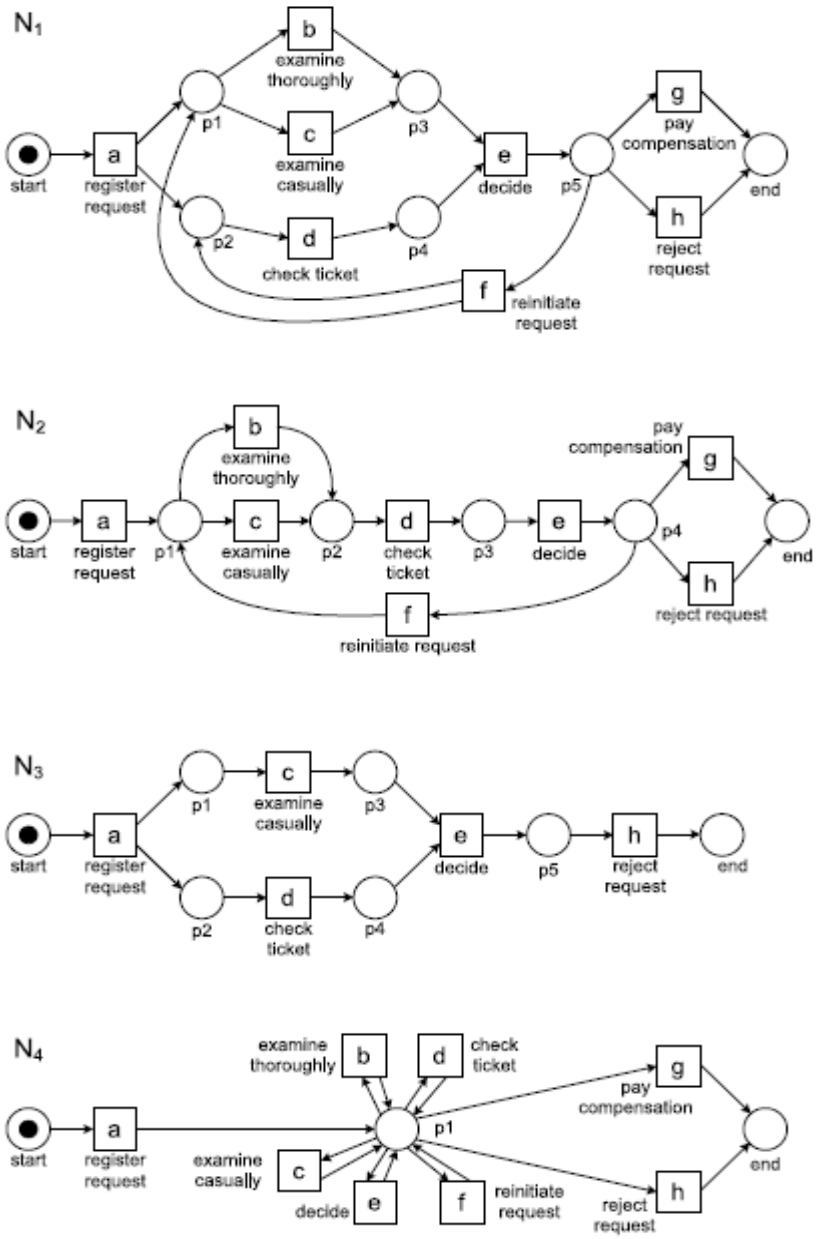

Figura 2.2 Le quattro WF-nets N_1, N_2, N_3 e N_4.

σ_1. Ad ogni fase sono associati quattro contatori: p (token prodotti), c (token consumati), m (token mancanti) e r (token rimanenti). Concentrandosi sui contatori p e c, inizialmente si presenta la situazione dove $p = c = 0$ e tutti i posti sono vuoti. In seguito, l'ambiente produce un token per il posto $start$. Pertanto, il contatore p è incrementato: $p = 1$. Considerata la traccia $\sigma_1 = (a, c, d, e, h)$, la transizione a è la prima ad essere eseguita. L'esecuzione di a consuma un token e ne produce due, per cui i contatori c e p sono incrementati rispettivamente di 1 e di 2 unità, ottenendo $c = 1$ e $p = 3$. Dopo la replicazione del secondo evento c, si ottengono $p = 4$ e $c = 2$. In seguito alla replicazione del terzo evento d, i contatori assumono i seguenti valori: $p = 5$ e $c = 3$. Replicando l'evento e, si consumano due token e se ne produce uno, da cui risulta $p = 6$ e $c = 5$. Dopo aver eseguito anche l'ultimo evento h, la situazione è la seguente: $p = 7$ e $c = 6$. Come ultimo passaggio, l'ambiente consuma un token dal posto finale end; il risultato finale è $p = c = 7$ e $m = r = 0$. Considerata l'assenza di token mancanti o rimanenti ($m = r = 0$), è evidente che non vi siano problemi a riprodurre la traccia σ_1. In generale, il fitness di un caso con traccia σ e WF-net N è definito come segue:

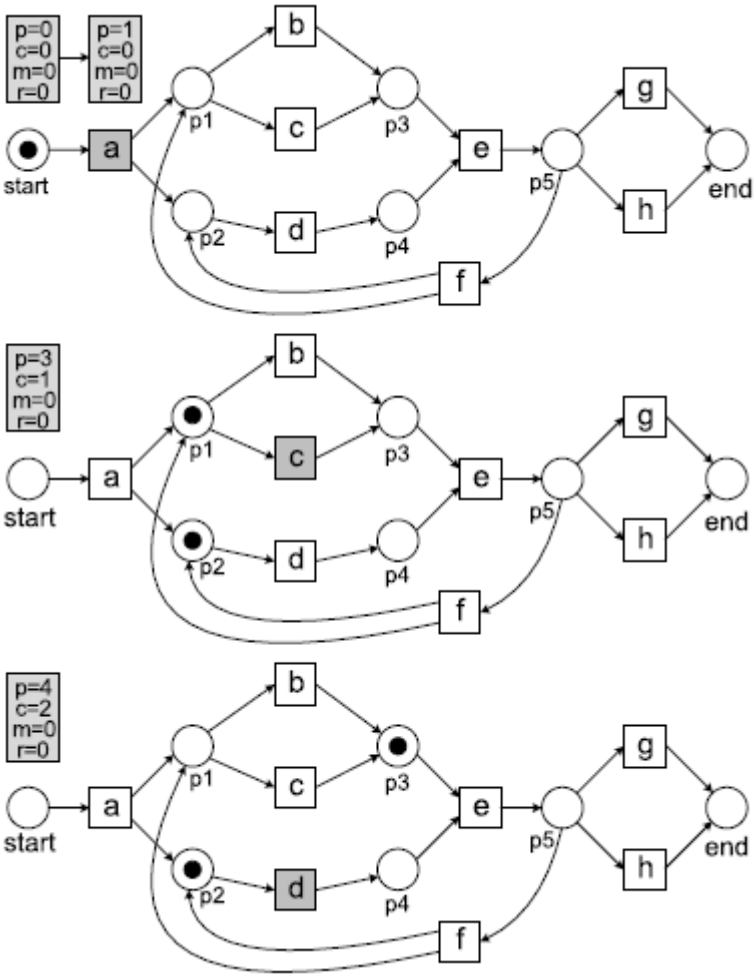

Figura 2.3a I primi tre passaggi relativi alla replicazione della traccia σ_1.

$$fitness(\sigma, N) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right)$$

Il primo termine calcola la frazione di token mancanti rispetto al numero di token consumati. In caso di nessun token mancante ($m = 0$), il termine $1 - \frac{m}{c}$ vale 1, mentre nel caso in cui tutti i token consumati fossero mancanti ($m = c$), il termine $1 - \frac{m}{c}$ vale 0. Analogamente, si ottiene $1 - \frac{r}{p} = 1$ come secondo termine nel caso in cui non vi siano token rimanenti e $1 - \frac{r}{p} = 0$ quando non si consuma nessuno dei token prodotti.

Per sua caratteristica, il fitness è definito tra valori compresi fra 0 e 1:

$$0 \leq fitness(\sigma, N) \leq 1$$

Nell'esempio di cui a pag. 40 , si ottiene il seguente valore di fitness:

$$fitness(\sigma_1, N_1) = \frac{1}{2}\left(1 - \frac{0}{7}\right) + \frac{1}{2}\left(1 - \frac{0}{7}\right) = 1$$

Il risultato vale 1 (fitting massimo) perché in tale esempio non vi sono token mancanti o rimanenti ($m = r = 0$). Si consideri ora il caso di una traccia non replicabile. Nella Figura 2.4 sono illustrati i diversi avanzamenti relativi alla riproduzione della traccia $\sigma_3 = (a, d, c, e, h)$ sulla rete workflow N_2. Come per l'esempio precedente, la situazione iniziale si presenta con $p = c = 0$ e con tutti i posti vuoti. L'ambiente produce il primo token per il posto iniziale $start$, incrementando il contatore p di 1 unità: $p = 1$. Dopo aver replicato l'evento a, i contatori aggiornati si presentano così: $p = 2,\ c = 1,\ m = 0$ e $r = 0$. Il secondo evento d non è replicabile, in quanto la transizione da a a d non è permessa dal modello N_2. Per renderla possibile,

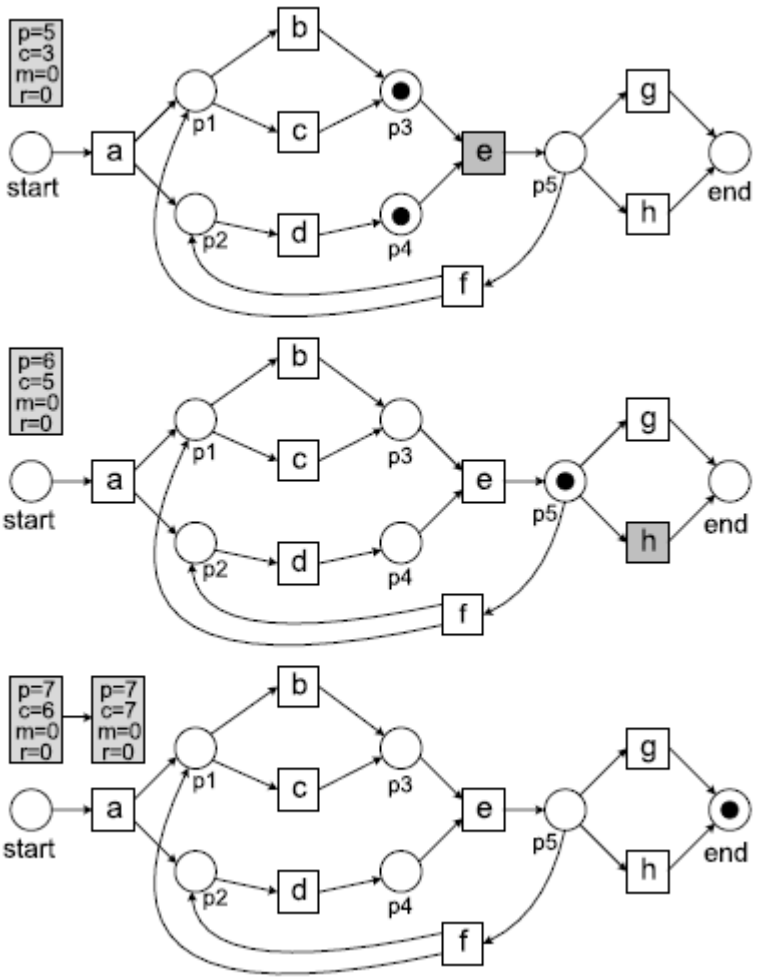

Figura 2.3b I restanti tre passaggi relativi alla replicazione della traccia σ_1.

è necessario aggiungere un token nel posto p_2; la registrazione del token mancante avviene incrementando il contatore m di 1 unità. Occorre inoltre applicare un'etichetta al posto p_2 al fine di ricordare la mancanza di un token. In seguito alla riproduzione di d, si ottengono $p = 3,\ c = 2,\ m = 1$ e $r = 0$. I restanti eventi (c, e, h) e le corrispondenti transizioni sono tutti consentiti, per cui sarà necessario solamente aggiornare i contatori p e c. Dopo aver replicato l'ultimo evento, la situazione è la seguente: $p = 6,\ c = 5,\ m = 1$ e $r = 0$. Nello stato finale $[p_2, end]$, l'ambiente consuma il token presente nel posto end, mentre, al contrario, rimane il token presente nel posto p_2. Ne consegue che sia il contatore c, sia il contatore r siano incrementati di 1 unità. Terminata la replicazione, il risultato finale è il seguente: $p = c = 6$ e $m = r = 1$. La Figura 2.4 fornisce anche una diagnosi utile a comprendere la natura della non conformità. Nel caso attuale, l'etichetta m indica una situazione in cui

si verifica l'evento d ma potrebbe non avvenire considerata la struttura del modello, mentre l'etichetta r indica una situazione in cui si suppone che l'evento d avvenga ma, osservando il log, non si verifica. Riguardo il fitness della traccia σ_3, la sua quantificazione è basata sui valori riscontrati di p, c, m e r:

$$fitness(\sigma_3, N_2) = \frac{1}{2}\left(1 - \frac{1}{6}\right) + \frac{1}{2}\left(1 - \frac{1}{6}\right) = 0.8333$$

Un terzo esempio è rappresentato dalla replicazione della traccia $\sigma_2 = (a, b, d, e, g)$ sulla rete workflow N_3; tale rete non contiene tutte le attività presenti nel log di eventi. In questo tipo di situazioni è ragionevole considerare solo gli eventi descrivibili dal modello; pertanto, si procede con la riproduzione della traccia $\sigma_2' = (a, d, e)$, il cui processo è mostrato in Figura 2.5. Il primo problema si origina con la replicazione dell'evento e; considerando che l'evento c non è presente nella traccia σ_2', il posto p_3 resta vuoto e la transizione e non può avvenire (regola di scatto non soddisfatta). Si fa fronte a tale problematica con l'inserimento di un'etichetta m sul posto p_3, in modo da registrare il token mancante ($m = 1$) e consentire la transizione e. Alla fine della riproduzione, la marcatura assunta dai token è $[p_1, p_5]$, per cui i token rimanenti sono due ($r = 2$); ai posti p_1 e p_5 si applica l'etichetta r. Come per i casi precedenti, l'ambiente necessita di consumare il token del posto finale end; tuttavia, ciò non è possibile in quanto il posto end non risulta "marcato". Infatti, come si può evincere dalla Figura 2.5, il posto end è vuoto. Per tale ragione, si registra un altro token mancante ($m = 2$) e si pone l'etichetta m

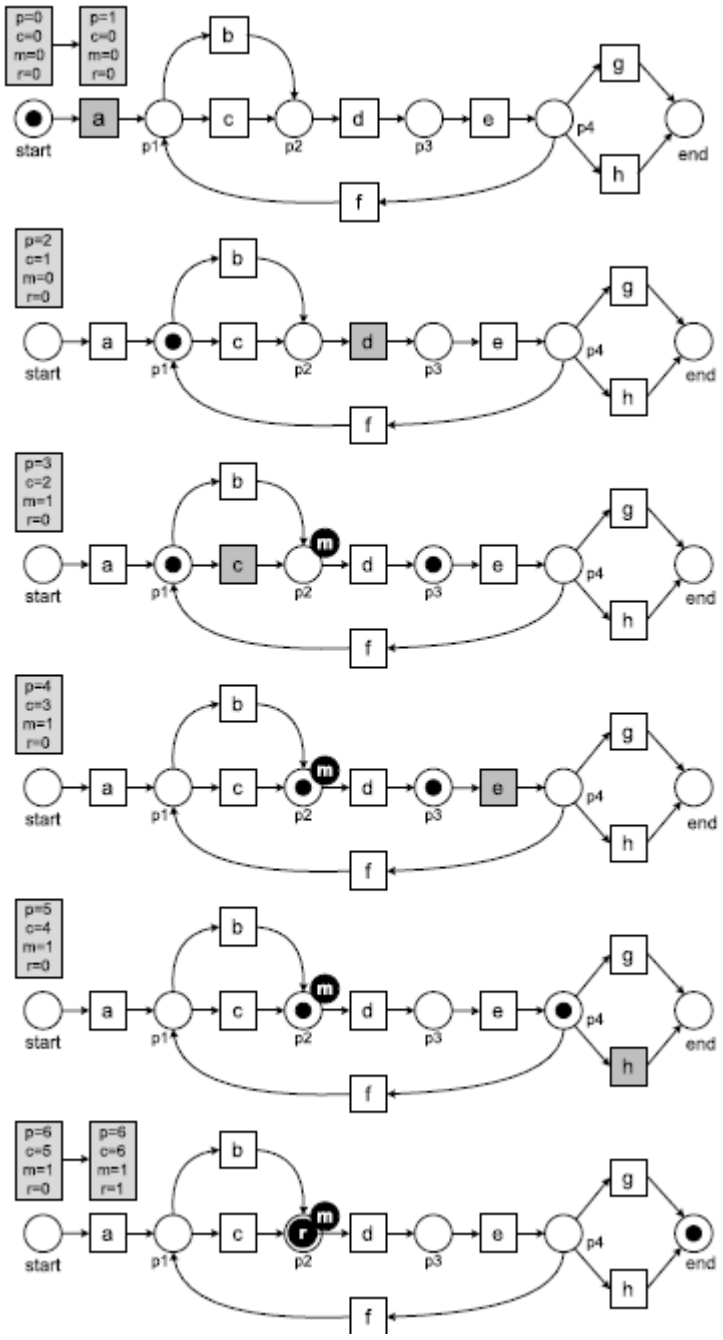

Figura 2.4 Il processo di replicazione della traccia σ_3.

sul posto end. Considerati i contatori $p = c = 5$ e $m = c = 2$, si ottiene il seguente valore di fitness per la traccia σ_2:

$$fitness(\sigma_2, N_3) = \frac{1}{2}\left(1 - \frac{2}{5}\right) + \frac{1}{2}\left(1 - \frac{2}{5}\right) = 0.6$$

La Figura 2.5, inoltre, evidenzia le cause di un livello di fitness così poco soddisfacente: gli eventi c ed h si presume che avvengano considerata la struttura del modello, ma non si verificano. L'evento e, invece, avviene anche se, guardando il modello, non sarebbe possibile. Le Figure 2.3a, 2.3b, 2.4 e 2.5 illustrano come analizzare il livello di fitness di un singolo caso; il medesimo approccio può essere utilizzato per analizzare il livello di fitness di un log di eventi costituito da molteplici casi. È sufficiente calcolare le somme di tutti i token prodotti, consumati, mancanti e rimanenti e applicare la formula precedente. Indicando con $p_{N,\sigma}$ il numero di token prodotti in seguito alla replicazione della traccia σ sul modello N, in accordo con la notazione fin qui utilizzata si denotano con $c_{N,\sigma}$, $m_{N,\sigma}$ e $r_{N,\sigma}$ le altre tre tipologie di token. Si può ora definire il fitness di un certo log L su una determinata rete workflow N:

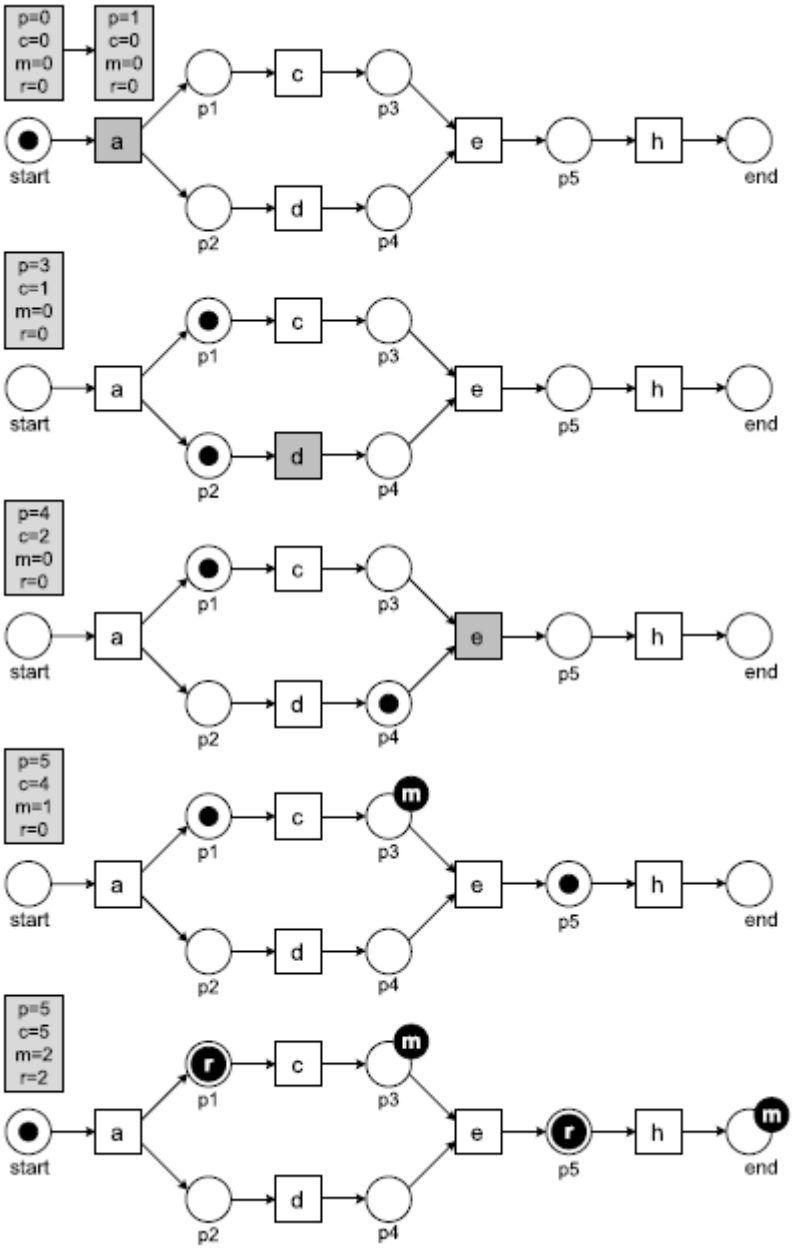

Figura 2.5 Il processo di replicazione della traccia σ_2.

$$fitness(L, N) = \frac{1}{2}\left(1 - \frac{\sum L(\sigma) \times m_{N,\sigma}}{\sum L(\sigma) \times c_{N,\sigma}}\right) + \frac{1}{2}\left(1 - \frac{\sum L(\sigma) \times r_{N,\sigma}}{\sum L(\sigma) \times p_{N,\sigma}}\right)$$

Si noti che il termine $\sum L(\sigma) \times m_{N,\sigma}$ rappresenta il numero totale di token mancanti in seguito all'intera replicazione del log; infatti, $L(\sigma)$ è la frequenza di una generica traccia σ, mentre $m_{N,\sigma}$ è il numero di

token mancanti per una singola istanza di σ. Lo stesso discorso, con i corrispondenti tipi di token, è valido per le altre tre sommatorie. Nonostante la formula $fitness(L, N)$ si concentri sui token e sui posti della rete, può essere interpretata come una misura riferita agli eventi. Ad esempio, il significato derivante da $fitness(L, N) = 0.9$ consiste nell'affermare che circa il 90% degli eventi presenti nel log L può essere correttamente replicato dalla rete workflow N. Sulla base delle informazioni e delle notazioni sopra illustrate, è possibile calcolare il livello di fitness riguardante il log di eventi L_{full}, replicato sulle quattro WF-nets N_1, N_2, N_3 e N_4 di Figura 2.2. I risultati sono i seguenti:

$$fitness(L_{full}, N_1) = 1$$
$$fitness(L_{full}, N_2) = 0.9504$$
$$fitness(L_{full}, N_3) = 0.8797$$
$$fitness(L_{full}, N_4) = 1$$

Come già anticipato, si evince che le WF-nets N_1 e N_4 sono in grado di replicare il log L_{full} senza riscontrare alcun tipo di problema. Per quanto riguarda $fitness(L_{full}, N_2) = 0.9504$, intuitivamente circa il 95% degli eventi contenuti in L_{full} può essere replicato sulla rete workflow N_2. Tale risultato, come spiegato in precedenza, può essere tradotto secondo due punti di vista:

- Il log L_{full} presenta un fitness pari a 0.9504, cioè il 5% degli eventi devia dal normale comportamento.
- Il modello di processo N_2 presenta un fitness pari a 0.9504, ossia il modello è incapace di spiegare il 5% del comportamento osservato nel log.

La prima visione si usa quando il modello in esame è considerato come normativo e corretto; la seconda visione si usa quando il modello in esame è descrittivo.

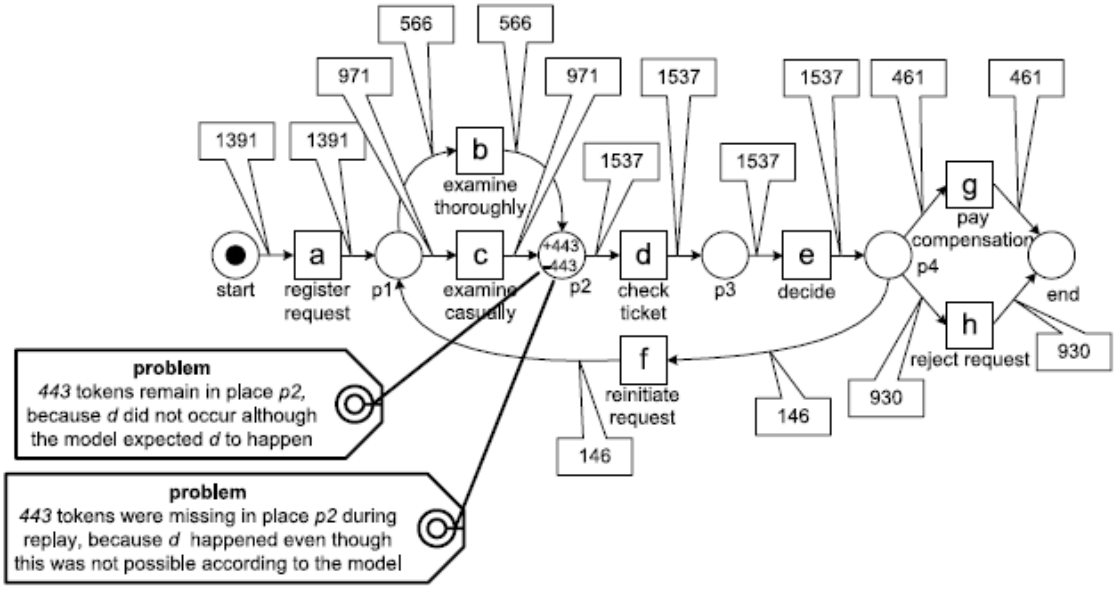

Figura 2.6 La diagnosi relativa a $fitness(L_{full}, N_2) = 0.9504$ con la quale si evidenziano le deviazioni.

La rete workflow N_3 presenta il più basso livello di fitness (0.8797). Tipicamente, il fitness basato sugli eventi è più alto rispetto all'approccio semplice e rapido; infatti, le reti N_2 e N_3 presentavano, con l'approccio semplice, fitness rispettivamente pari a 0.6815 e 0.4543, contro gli attuali 0.9504 e 0.8797. La Figura 2.6 mostra alcune diagnosi generate dalla replicazione del log di eventi L_{full} sul modello N_2. I numeri presenti sugli archi della rete indicano il flusso dei token prodotti e consumati; da essi si può evincere il percorso compiuto dai casi analizzati lungo il modello. Ad esempio, in Figura 2.6 sono presenti 930 richieste rigettate e 461 richieste accolte. Inoltre, i posti ai quali erano state applicate le etichette m e r possono essere aggregati per individuare problemi di conformità e rilevarne la gravità. Come si può evincere dalla Figura 2.6, per 443 volte l'evento d è occorso malgrado non fosse supposto il suo avvenimento, mentre per altre 443 volte l'evento d non si è verificato nonostante il suo avvenimento fosse supposto. La causa è da ricondurre all'esecuzione dell'evento d avvenuta in antecedenza all'evento b o c, condizioni entrambe impossibili considerando la struttura sequenziale del modello N_2. Analogamente, la Figura 2.7 illustra le informazioni diagnostiche relative alla rete N_3. I problemi rilevati su tale rete sono più severi dei precedenti riscontrati su N_2; per esempio, in 566 occasioni l'attività *e* è occorsa senza che prima si fosse verificata l'attività c, oppure ancora 461 casi non hanno raggiunto il posto finale end perché la richiesta non è stata rigettata (attività h).

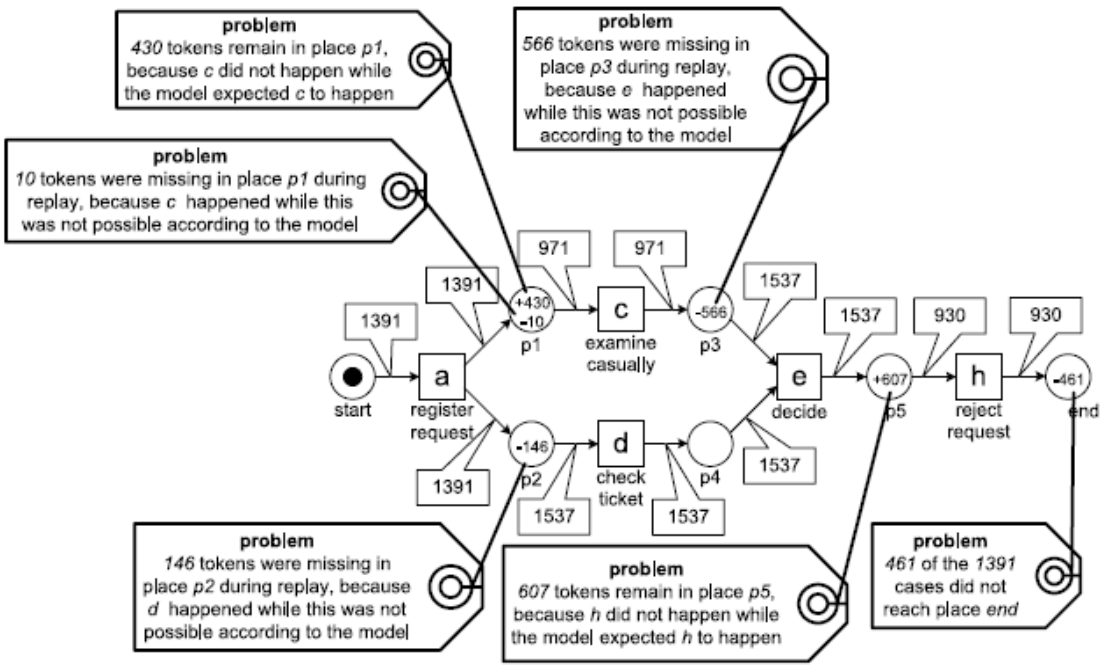

Figura 2.7 La diagnosi relativa a $fitness(L_{full}, N_3) = 0.8797$ *con la quale si evidenziano le deviazioni.*

Un log di eventi, come si può vedere in Figura 2.8, può essere suddiviso in due parti: un log di eventi contenente solo i casi conformi (fitting cases) e un log di eventi contenente solo i casi non conformi (non-fitting cases). Ciascuno dei due log di eventi può essere utilizzato per ulteriori analisi; ad esempio, si potrebbe costruire un modello destinato a log di eventi costituiti solamente da casi devianti e non

conformi. Un altro esempio riguarda le informazioni secondarie o aggiuntive; potrebbe essere d'interesse risalire a quali persone compete la gestione dei casi devianti, se tali casi richiedono molto tempo per essere evasi o se generano costi rilevanti.

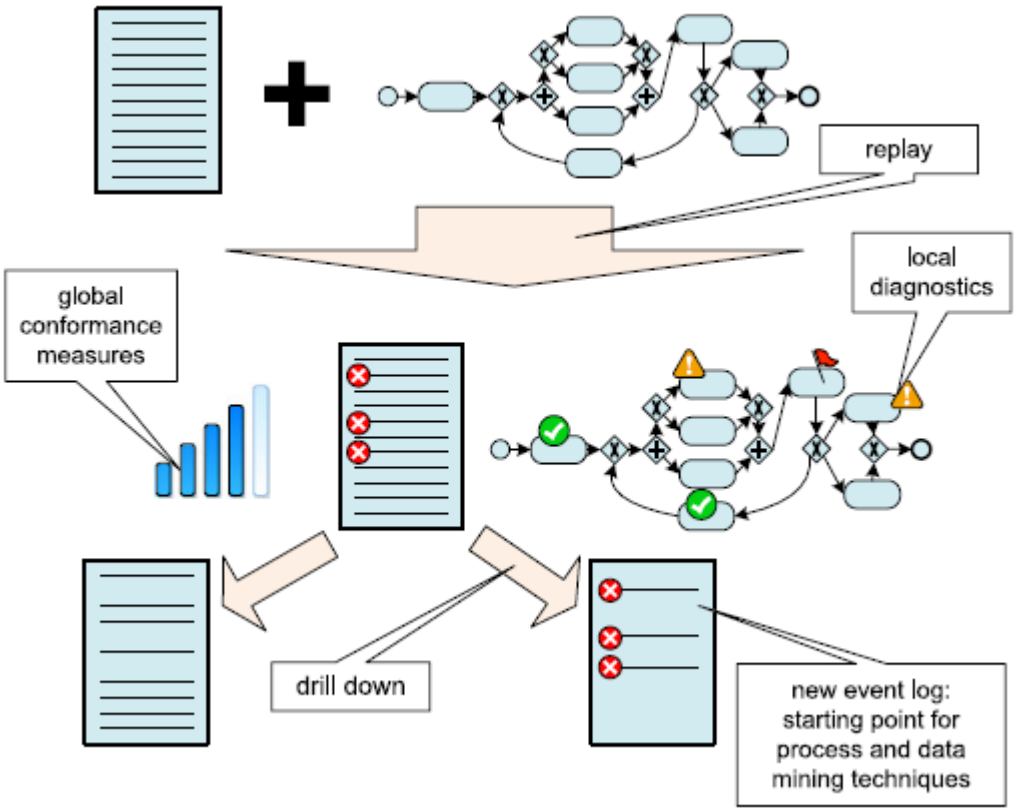

Figura 2.8 Scomposizione di un log rispetto ai casi conformi e non conformi, dando origine a due nuovi log di dimensioni minori.

È bene precisare che la replicazione dei log sui modelli di processo non si limita alle reti di Petri; qualsiasi altra notazione avente una semantica eseguibile e rappresentabile è adatta alla replicazione dei log. Tuttavia, la presenza nelle WF-nets di posti chiari ed espliciti, soprattutto i posti d'inizio e di fine, facilita la generazione di diagnosi significative e veritiere. Le tecniche di replicazione dei log sono fruibili anche per analisi relative alla precisione (underfitting) e alla generalizzazione (overfitting) dei modelli; è possibile condurre tali analisi tenendo traccia del numero di transizioni consentite nel corso della replicazione. Se, mediamente, sono molte le transizioni abilitate durante la riproduzione di un log, è probabile che il modello soffra di underfitting; al contrario, in caso di numero ridotto di transizioni consentite, è probabile che il modello sia affetto da overfitting. Ad esempio, nel modello N_4 le attività b, c, d, e e f sono costantemente consentite; una tale situazione suggerisce che il modello N_4 sia affetto da underfitting.

Completata la replicazione del log, si procede con la comparazione dei footprint, già precedentemente definiti come le matrici rappresentanti le dipendenze causali sussistenti fra le attività del log in esame. Il footprint relativo al log di eventi L_{full} è mostrato in Tabella 2.2. Oltre ai log di eventi, anche i modelli di processo hanno un loro footprint; molto semplicemente, è sufficiente generare un log di eventi

completo, ossia si effettua il play-out e si registrano le sequenze eseguite. Si ricorda che, in merito al footprint, un log di eventi è completo se e solo se tutte le attività che possono seguire un'altra attività si comportano in tale maniera almeno una volta all'interno del log. Applicando tale definizione alla rete workflow N_1 di Figura 2.2, si ottiene esattamente il footprint di Tabella 2.2. Il significato risultante suggerisce che il log di eventi e il modello sono conformi l'uno rispetto all'altro.

Tabella 2.2 Il footprint di L_{full} e N_1.

	a	b	c	d	e	f	g	h
a	#	→	→	→	#	#	#	#
b	←	#	#	||	→	←	#	#
c	←	#	#	||	→	←	#	#
d	←	||	||	#	→	←	#	#
e	#	←	←	←	#	→	→	→
f	#	→	→	→	←	#	#	#
g	#	#	#	#	←	#	#	#
h	#	#	#	#	←	#	#	#

In Tabella 2.3 è invece riportato il footprint della rete workflow N_2. Comparando i due footprint di Tabella 2.2 e Tabella 2.3 si riscontrano delle differenze, riassunte in Tabella 2.4.

Tabella 2.3 Il footprint di N_2.

	a	b	c	d	e	f	g	h
a	#	→	→	#	#	#	#	#
b	←	#	#	→	#	←	#	#
c	←	#	#	→	#	←	#	#
d	#	←	←	#	→	#	#	#
e	#	#	#	←	#	→	→	→
f	#	→	→	#	←	#	#	#
g	#	#	#	#	←	#	#	#
h	#	#	#	#	←	#	#	#

Ad esempio, la relazione fra a e d cambia passando da → a #. Dalla comparazione del log di eventi L_{full} con la rete workflow N_2 si evince che, nel log, l'attività a è direttamente seguita dall'attività d, mentre nel modello N_2 tale soluzione non è possibile. La comparazione dei footprint è utile anche alla quantificazione della conformità; osservando la Tabella 2.4 si nota che 12 celle su 64 differiscono. Pertanto, si può affermare che il livello di conformità basato sui footprint sia pari a $1 - \frac{12}{64} = 0.8125$. Tale procedura è significativa solo in caso di log completo rispetto alla relazione di successione diretta $>_L$.

Tabella 2.4 Differenze fra i footprint di L_{full} e di N_2.

	a	b	c	d	e	f	g	h
a				→ : #				
b				|| : →	→ : #			
c				|| : →	→ : #			
d	← : #	|| : ←	|| : ←			← : #		
e		← : #	← : #					
f				→ : #				
g								
h								

Come detto, sia i modelli che i log di eventi sono descrivibili con i footprint. Tale analogia permette di eseguire comparazioni di tipo log-modello come sopra illustrato, ossia è possibile controllare se il log e il modello siano concordanti in merito all'ordinamento delle attività. Lo stesso approccio è valido anche per le comparazioni di tipo log-log e modello-modello. La comparazione dei footprint di due modelli (modello-modello) consente la quantificazione della similarità. La comparazione dei footprint di due log di eventi (log-log) può essere usata, ad esempio, per rilevare il concept drift. Tale termine si riferisce alla situazione in cui il processo cambia in corso di analisi. Per esempio, un log di eventi inizialmente presenta due attività parallele che, successivamente, diventano sequenziali. Una situazione di questo tipo può essere scoperta e investigata dividendo il log in log più piccoli, dei quali si analizzano i footprint. La comparazione log-log di una sequenza di eventi può rivelare la presenza di un concept drift.

Il conformance checking, oltre che per identificare le differenze esistenti tra il modello di processo e il processo reale descritto nel log di eventi, può essere usato per altri scopi.

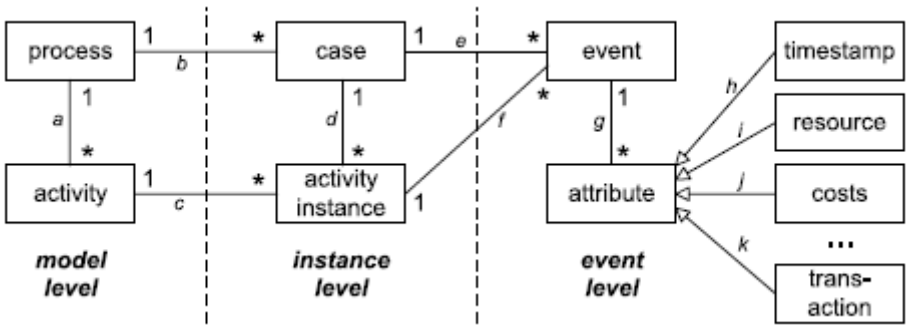

Figura 2.9 I tre livelli che si originano dalla connessione fra il log di eventi e il modello di processo.

Una prima destinazione è la riparazione dei modelli, ossia si cerca di allineare il modello alla realtà considerata la loro iniziale discordanza. Le Figure 2.6 e 2.7, ad esempio, possono essere di grande aiuto alla riparazione del modello; in base ai valori di frequenza, si può decidere di eliminare le attività di rado o per nulla eseguite. Le informazioni relative alle etichette m e r, cosi come i footprint, sono

funzionali per il progettista al fine di riparare il modello. Un'altra destinazione d'uso per il conformance checking riguarda il collegamento che si stabilisce fra log di eventi e modello di processo. Mettendo in relazione gli eventi con le attività, è possibile estrarre informazioni dal log con le quali arricchire il processo; per esempio, data e ora registrati in un log possono servire ad asserire con alta probabilità quali siano i tempi di durata di una determinata attività modellata. La Figura 2.9 pone in evidenza le connessioni che si instaurano fra il modello di processo e il log di eventi durante la replicazione. Come si può osservare, si definiscono tre differenti livelli: il livello modello, il livello istanza e il livello evento. Tipicamente, il livello modello e il livello evento esistono indipendentemente l'uno dall'altro; tale aspetto si traduce con un accoppiamento vago e approssimativo di eventi e modelli di processo. Fortunatamente, la replicazione del log è abile anche a stabilire un accoppiamento più stretto e consistente fra i livelli modello e evento. Il livello istanza è costituito da casi e istanze di attività che legano i processi e le attività presenti nel modello con gli eventi contenuti nel log. Ogni evento richiede la correlazione con un caso; in fase di replicazione di un log di eventi su un modello, ogni evento che ben si conforma al modello è connesso ad un'istanza di attività. Si noti che un singolo caso può contenere diverse istanze della stessa attività; inoltre, ad una singola istanza di attività possono corrispondere molteplici eventi. A valle di tali considerazioni, risulta che tutte le informazioni estratte dal log di eventi possono essere proiettate sul modello, ad esempio per rilevare i colli di bottiglia o per evidenziare i cammini a maggiore frequenza.

3. Supporto operativo e process enhancement

In principio, il process mining analizzava eventi appartenenti a casi che erano già stati eseguiti e completati; perciò, le tecniche si applicavano in modalità offline e consistevano in una sorta di analisi a posteriori. Attualmente, è possibile condurre un'analisi di process mining anche online, con la quale è possibile fornire al processo un supporto operativo in tempo reale. Considerando le varie distinzioni fra dati storici (post-mortem) e dati attuali (pre-mortem) e fra modelli di fatto (de facto models) e modelli di diritto (de jure models), si può definire la struttura raffinata del process mining[48], illustrata in Figura 3.1. Per convenzione, si assume che i modelli di processo, le persone, le organizzazioni rappresentino il "mondo esterno"; il sistema informativo registra le informazioni inerenti il mondo esterno in maniera tale da permettere l'estrazione di dati dai log di eventi. Per enfatizzare la sistematica e affidabile registrazione degli eventi, la Figura 3.1 utilizza il vocabolo provenienza (provenance).

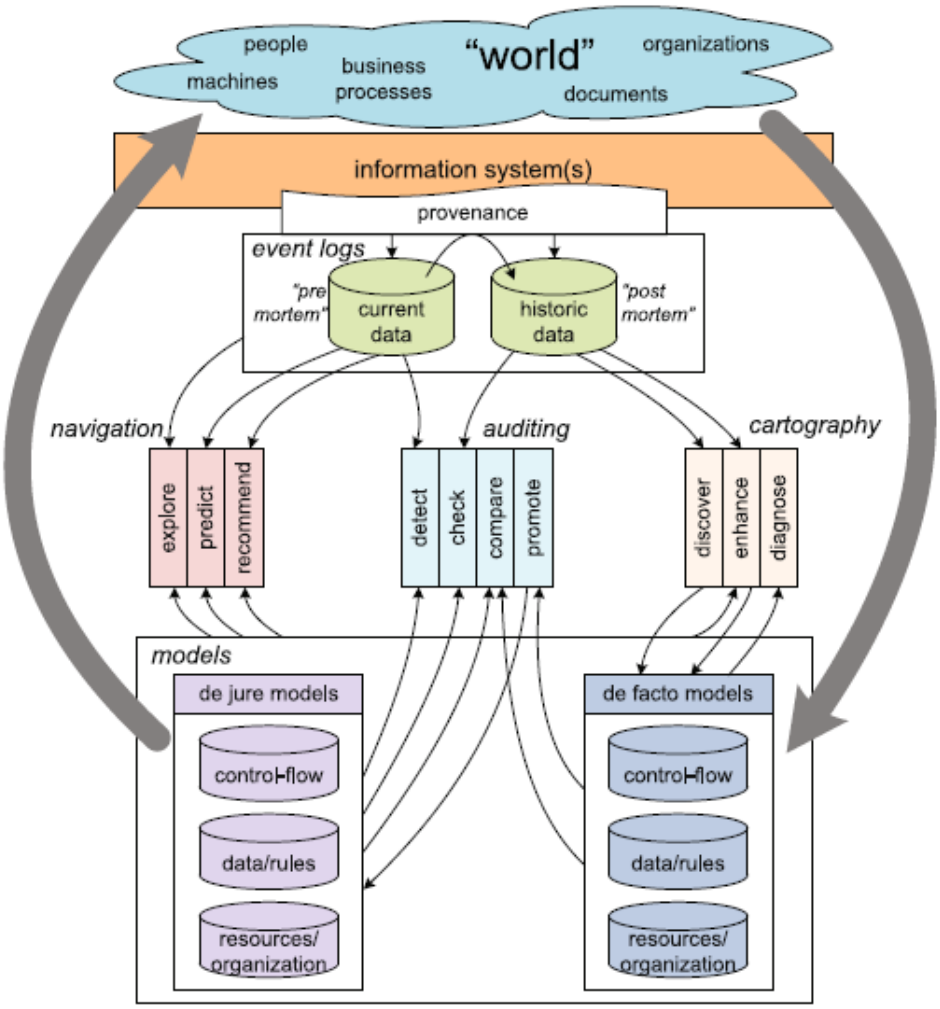

Figura 3.1 La struttura raffinata del process mining.

[48] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, p. 241.

Tale termine deriva dalle scienze computazionali, dove lo si usa in riferimento ai dati necessari per rendere possibile la riproduzione di un esperimento. Il business process provenance[49] punta a raccogliere sistematicamente le informazioni necessarie a ricostruire ciò che è realmente avvenuto in un processo o in un'organizzazione. È opportuno assicurarsi che i log di eventi non siano alterati e che sappiano descrivere adeguatamente e fedelmente la "storia" di un processo. Pertanto, il business process provenance si riferisce all'insieme di attività essenziali ad assicurare che la storia, così come è descritta nei log, non possa essere oscurata o riscritta, e serva come base fondamentale per il controllo e il miglioramento del processo. Come anticipato, la struttura raffinata del process mining discerne i log di eventi nelle categorie post mortem e pre mortem[50]; la prima tipologia si riferisce ai dati inerenti le informazioni sui casi che sono stati completati, ossia possono essere usati a fini di controllo e miglioramento del processo, ma non sono in grado di influenzare i casi ai quali si riferiscono. I log di eventi sin qui presi in considerazione contengono solamente dati storici, cioè del tipo post mortem. I dati pre mortem, invece, si riferiscono ai casi non ancora completati. Se un caso è in fase di esecuzione, ossia è ancora "vivo" (pre mortem), potrebbe essere possibile sfruttare l'informazione contenuta nel log riguardante tale caso allo scopo di garantire una gestione corretta ed efficiente del caso stesso. I dati post mortem sono di grande rilevanza per il process mining condotto in modalità offline; ad esempio, si può tentare di scoprire il flusso di controllo di un processo basandosi su una serie storica di dati avente ampiezza di un anno. Per il process mining in modalità online è necessario disporre di un misto di dati pre mortem (attuali) e dati post mortem (storici). Per esempio, le informazioni storiche possono essere utili ad apprendere un modello predittivo; in seguito, si combinano le informazioni relative al caso in esecuzione con il modello predittivo, in modo tale da ricavare una stima del tempo di flusso rimanente del caso medesimo. La distinzione avviene anche per i modelli, i quali sono raggruppati in due categorie: de jure models e de facto models. Il primo gruppo comprende modelli normativi, ossia modelli che spiegano quali azioni intraprendere e come svolgerle. Ad esempio, un modello di processo usato per configurare un sistema BPM è normativo e obbliga le persone a esercitare le proprie mansioni secondo particolari modalità. Un modello de facto, invece, è di tipo descrittivo e non si pone come obiettivo il governo o il controllo della realtà. Le tecniche di process discovery illustrate nel Capitolo 1 danno origine a modelli di tale tipologia. Come si può osservare dalla Figura 3.1, i modelli di fatto (de facto) sono derivati dalla realtà, mentre i modelli di diritto (de jure) si prefiggono di riuscire a influenzare la realtà. Inoltre, si definiscono dieci attività relative al process mining, raggruppate in tre categorie: cartografia, auditing e navigazione. Partendo dalla cartografia, i modelli possono essere assimilati a mappe che descrivono il processo, cercando di rappresentare la realtà.

[49] W.M.P. van der Aalst, op. cit, p. 242.
[50] W.M.P. van der Aalst, op. cit, p. 243.

Le tre attività racchiuse nella cartografia[51] sono:

- Scoprire. Si riferisce alle tecniche di process discovery precedentemente discusse.
- Migliorare. Legando i modelli di processo con i log di eventi è possibile migliorare i modelli stessi, in modo che siano adeguatamente allineati ai comportamenti reali.
- Diagnosticare. Tale attività non utilizza direttamente i log di eventi e si concentra sulle analisi di processi basati sui modelli.

L'auditing è l'insieme di attività volte a controllare se i processi operativi sono eseguiti entro certi confini stabiliti dal management. Le attività costituenti l'auditing[52] sono le seguenti:

- Rilevare. Consiste nella comparazione fra modelli normativi (de jure) e dati attuali (pre mortem) al fine di individuare eventuali deviazioni in tempo reale. Nel momento in cui una regola risulta violata, si genera un segnale di allarme.
- Controllare. Lo scopo di tale attività è quantificare il livello di conformità del modello in seguito al rilevamento di deviazioni.
- Comparare. I modelli di fatto possono essere confrontati con i modelli di diritto per osservare come e in che misura la realtà si discosti dai comportamenti attesi e pianificati. Per tale scopo, non si utilizzano in modo diretto i log di eventi; come illustrato in precedenza, i footprint sono utilizzabili per le attività di comparazione.
- Promuovere. Basandosi sulla precedente comparazione e considerando il valore di fitting, è possibile promuovere parti del modello di fatto in un nuovo modello di diritto; così facendo, i processi possono essere affinati.

Si noti che le prime due attività sono del tutto simili, tranne che per i dati utilizzati: la rilevazione è condotta in modalità online utilizzando dati di tipo pre mortem, al fine di intervenire tempestivamente all'insorgere di una discrepanza, mentre il controllo utilizza dati di tipo post mortem ed è eseguito in modalità offline. L'ultima delle tre categorie è la navigazione, la quale, a differenza della cartografia e dell'auditing, si proietta in avanti e guarda al futuro. Ad esempio, si potrebbero adoperare delle tecniche di process mining al fine di prevedere il futuro di un particolare caso e coadiuvare l'utente nella scelta delle azioni più adatte. Le tre attività costituenti la navigazione[53] sono di seguito elencate:

- Esplorare. La combinazione di dati relativi agli eventi con i modelli è funzionale all'esplorazione dei processi in tempo reale. I casi in esecuzione possono essere visualizzati e confrontati con altri casi simili affrontati precedentemente.
- Prevedere. Combinando le informazioni associate ai casi in esecuzione con i modelli, è possibile realizzare previsioni circe le situazioni future, come ad esempio il tempo di flusso rimanente.

[51] W.M.P. van der Aalst, op. cit, p. 244.
[52] W.M.P. van der Aalst, op. cit, pp. 244-245.
[53] W.M.P. van der Aalst, op. cit, p. 245.

- Consigliare. Le informazioni utilizzate per costruire le previsioni sono utili anche a suggerire e raccomandare le azioni più adatte da intraprendere.

Come già detto, tradizionalmente il process mining operava in modalità offline; di conseguenza, erano presi in considerazione solo i dati di tipo post mortem. Il significato di tale procedimento consiste nel considerare per le analisi di process mining solo casi completati. Tuttavia, per avvalersi del supporto operativo è necessario operare in modalità online e considerare anche i dati di tipo pre mortem; ciò significa che sono da considerare solo i casi in esecuzione, in quanto, potenzialmente, possono essere ancora influenzati.

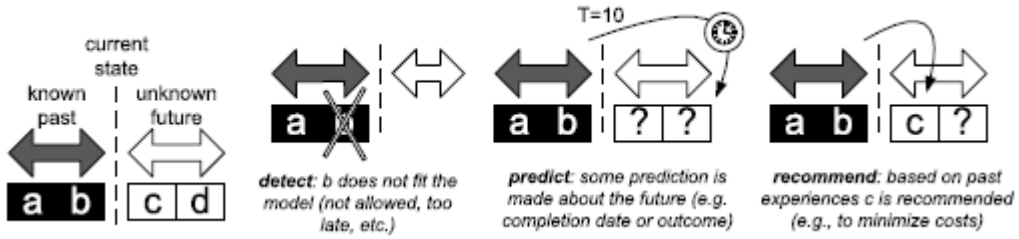

Figura 3.2 Le tre attività del process mining relative al supporto operativo.

La Figura 3.2 sintetizza l'essenza del supporto operativo. Si considerino, ad esempio, due attività a e b da eseguire; la traccia parziale $\sigma_p = (a, b)$ descrive la storia passata del caso. Diversamente, il futuro del caso in esame, dopo aver osservato σ_p, non è ancora noto. Una possibile ipotesi prevede anche l'esecuzione delle attività c e d, da cui risulta la traccia completa $\sigma_c = (a, b, c, d)$. In Figura 3.2 sono mostrate le attività relative al supporto operativo: rilevare (detect), prevedere (predict) e consigliare (recommend). Tali attività, già declinate in riferimento alla Figura 3.1, ricorrono ad alcune assunzioni: ad esempio, le previsioni e i suggerimenti potrebbero basarsi su dei modelli di regressione o essere ricavati da tecniche di simulazione. Inoltre, le tecniche di process mining possono essere modificate in modo da fornire supporto operativo. A tale scopo, si consideri il log di eventi illustrato in Tabella 3.1.

Tabella 3.1 Frammento di un log di eventi contenente informazioni sulle coordinate temporali e sulle transizioni. Ad esempio, l'evento a^{12}_{start} indica l'inizio dell'attività a all'istante 12.

ID caso	Traccia
1	$(a^{12}_{start}, a^{19}_{complete}, b^{25}_{start}, d^{26}_{start}, b^{32}_{complete}, d^{33}_{complete}, e^{35}_{start}, e^{40}_{complete}, h^{50}_{start}, h^{54}_{complete})$
2	$(a^{17}_{start}, a^{23}_{complete}, d^{28}_{start}, c^{30}_{start}, d^{32}_{complete}, c^{38}_{complete}, e^{50}_{start}, e^{59}_{complete}, g^{70}_{start}, g^{73}_{complete})$
3	$(a^{25}_{start}, a^{30}_{complete}, c^{32}_{start}, c^{35}_{complete}, d^{35}_{start}, d^{40}_{complete}, e^{45}_{start}, e^{50}_{complete}, f^{50}_{start}, f^{55}_{complete}, b^{60}_{start}, d^{62}_{start}, b^{65}_{complete}, d^{67}_{complete}, e^{80}_{start}, e^{87}_{complete}, g^{90}_{start}, g^{98}_{complete})$
…	…

Il modello che descrive il processo i cui eventi sono contenuti in Tabella 3.1 può essere riprodotto mediante varie notazioni; per esempio, in Figura 3.3, è illustrato il sistema di transizione che modella

tale processo. Il sistema di transizione etichetta ciascun nodo con la marcatura della corrispondente rete di Petri; ogni attività è modellata con una transizione di inizio ($start$) e una transizione di fine ($complete$). Ad esempio, la transizione a_{start} consuma un token dal posto $start$ e ne produce un altro nel posto a, il quale indica che l'attività a è in esecuzione. La transizione $a_{complete}$ consuma un token dal posto a e ne produce uno per ciascuno dei suoi dei suoi posti in uscita, p_1 e p_2 (marcatura $[p_1, p_2]$). Un altro esempio è la marcatura $[b, d]$, che indica l'esecuzione parallela delle attività b e d.

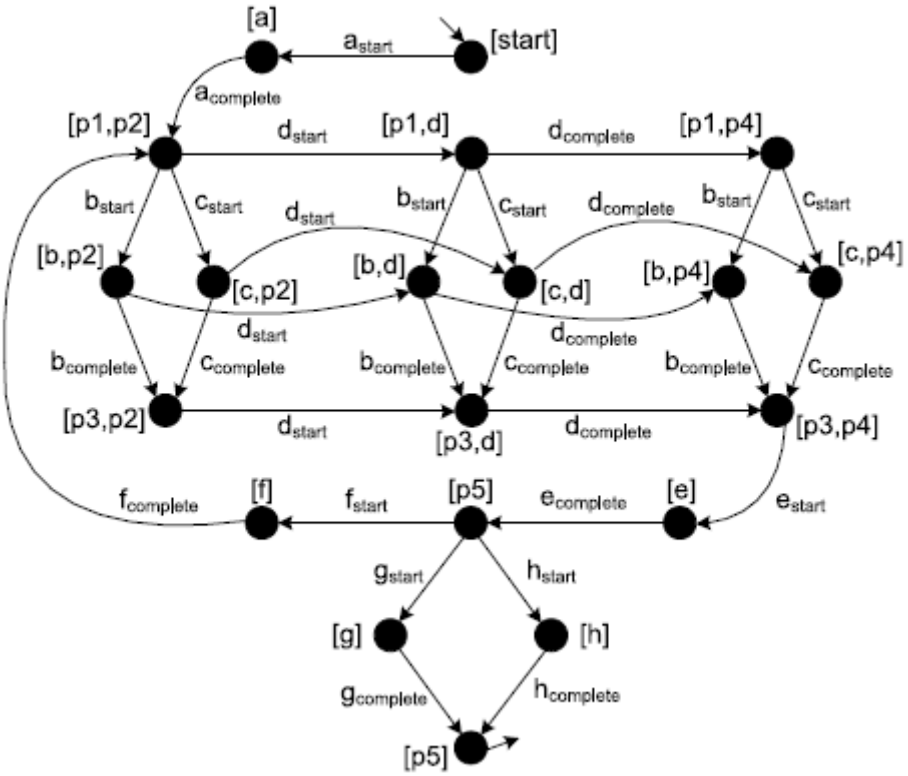

Figura 3.3 Sistema di transizione modellante il processo a cui si riferisce il log di eventi mostrato in Tabella 3.1.

Come spiegato in precedenza, il supporto operativo è costituito da tre attività; la prima attività consiste nel rilevare le deviazioni in tempo reale[54]. Tale attività può essere vista come una sorta di conformance checking dinamico, ma che presenta due grandi differenze rispetto al conformance checking descritto nel presente elaborato: non si considerano i log interi ma bensì tracce parziali di un determinato caso e, al manifestarsi di una deviazione, deve seguire un'immediata reazione.

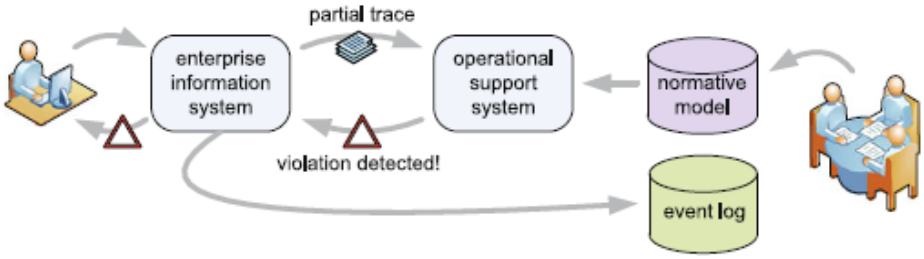

Figura 3.4 Schema riassuntivo di come si rilevano le deviazioni.

[54] W.M.P. van der Aalst, op. cit, pp. 247-249.

La Figura 3.4 illustra tale tipologia di supporto operativo. Gli utenti interagiscono con il sistema informativo aziendale e gli eventi sono registrati in base alle loro azioni. La traccia parziale di ciascun caso analizzato è continuamente controllata dal sistema, il quale fornisce supporto operativo; i controlli avvengono ogniqualvolta occorra un evento. Il sistema genera istantaneamente un segnale di allarme in caso di deviazione rilevata; in seguito a tale allarme, il sistema informativo e i suoi utenti possono adottare le azioni più appropriate. Tutti i casi del log di eventi di Tabella 3.1 sono conformi al sistema di transizione di Figura 3.3; pertanto, in fase di esecuzione di tali casi, non sarà rilevata alcuna deviazione.

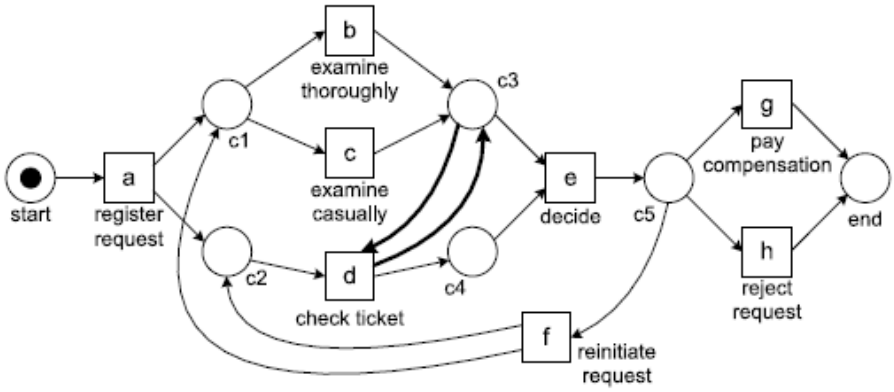

Figura 3.5 La rete workflow N_1 con l'aggiunta di un ulteriore vincolo sull'attività d.

Si consideri ora la più restrittiva rete workflow di Figura 3.5 descrivente il comportamento normativo che si desidera ottenere dal modello. Come si può osservare, l'attività d non può iniziare prima che l'attività b o c sia stata completata, vincolo non presente nel modello originale N_1 di Figura 2.2. Considerando il primo dei tre casi contenuti nel log di eventi di Tabella 3.1, ossia $\sigma_1 = \left(a^{12}_{start}, a^{19}_{complete}, b^{25}_{start}, d^{26}_{start}, b^{32}_{complete}, d^{33}_{complete}, e^{35}_{start}, e^{40}_{complete}, h^{50}_{start}, h^{54}_{complete}\right)$, si nota che l'esecuzione dei primi tre eventi non presenta problemi, quindi non si rileva alcuna deviazione; la marcatura in seguito alla loro esecuzione è $[c_2, b]$. L'evento successivo, d^{26}_{start}, non può essere eseguito in presenza di tale marcatura. Pertanto, si genera un segnale di allarme all'istante 26 relativo alla traccia parziale $\left(a^{12}_{start}, a^{19}_{complete}, b^{25}_{start}, d^{26}_{start}\right)$; l'allarme segnala che l'attività d è iniziata senza essere stata prima abilitata. Per il secondo caso si rileva una deviazione all'istante 28, relativo alla traccia parziale $\left(a^{17}_{start}, a^{23}_{complete}, d^{28}_{start}\right)$. Il segnale di allarme si genera per la medesima ragione del precedente caso. Tale motivazione spiega anche il segnale di allarme che si genera nel terzo caso, in cui si registra una deviazione all'istante 62 relativa alla traccia parziale $(a^{25}_{start}, \dots, d^{62}_{start})$. Tali esempi mostrano che l'approccio basato sulla replicazione degli eventi, descritto nel capitolo precedente, può anche essere utilizzato per rilevare in tempo reale le deviazioni.

La seconda attività del supporto operativo è la previsione[55]. La Figura 3.6 presenta nuovamente gli utenti che interagiscono con il sistema informativo aziendale. Gli eventi, registrati caso per caso, possono essere inviati al sistema di supporto operativo nella forma di tracce parziali; sulla base di esse e con l'aiuto di un modello predittivo, si genera una previsione. Il tempo di flusso rimanente è di 10 giorni, il costo totale del caso è di 5000€ e la probabilità che un caso sia rigettato è pari a 0.75 sono esempi di previsione.

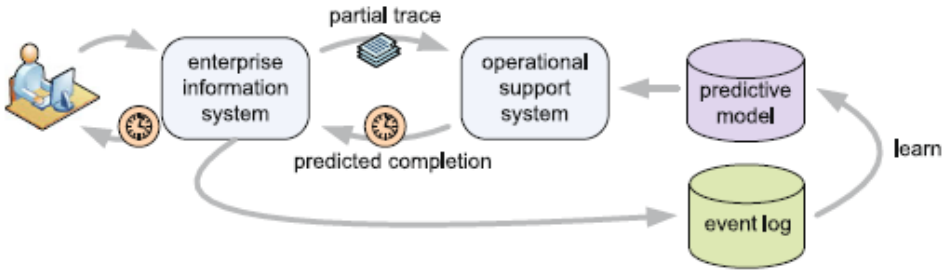

Figura 3.6 Sia le tracce parziali che i modelli predittivi contribuiscono alla generazione di previsioni.

Sono varie le tecniche adottabili per generare delle previsioni. Ad esempio, si possono estrarre dalle tracce parziali alcune proprietà importanti, tradotte in forma di variabili di risposta; tali variabili sono spesso degli indicatori di performance (es. tempo di flusso, costo totale, ecc.). Se la variabile di risposta è di tipo numerico, solitamente si utilizza l'analisi di regressione per costruire una previsione; nel caso in cui la variabile di risposta sia di tipo categorico, si predilige l'uso di tecniche di classificazione come, ad esempio, gli alberi di decisione. Il modello predittivo, seppur basato su dati storici, è fruibile al fine di generare previsioni riguardo i casi ancora in fase di esecuzione. Le tecniche di previsione presentano grande varietà e un ampio spettro di possibili domande; considerata la connotazione rivolta all'ambito produttivo del presente elaborato, è giusto porre l'attenzione su come prevedere il tempo di flusso rimanente di una certa attività. A tale scopo, si utilizzerà un sistema di transizione. Il punto di partenza per questo tipo di approccio è il log di eventi, corredato dalle coordinate temporali, della tabella 3.1. Assumendo che il log sia in fitting con il sistema di transizione, è possibile riprodurre gli eventi sul modello e raccogliere le informazioni temporali; gli eventi o i casi che mal si adattano al modello, possono essere ignorati o gestiti in altra maniera, secondo quanto descritto nel capitolo 2. La Figura 3.7 mostra le replicazioni, con i rispettivi riferimenti temporali, delle prime due tracce di Tabella 3.1. Si consideri il primo caso:

$$(a^{12}_{start}, a^{19}_{complete}, b^{25}_{start}, d^{26}_{start}, b^{32}_{complete}, d^{33}_{complete}, e^{35}_{start}, e^{40}_{complete}, h^{50}_{start}, h^{54}_{complete})$$

Tale caso inizia all'istante 12 e si conclude all'istante 54; pertanto, il tempo di flusso è di 42 unità temporali. Gli stati visitati dal caso sono descritti da un'etichetta (t, e, r, s), dove t è l'istante in cui si visita lo stato, e è il tempo trascorso dall'inizio sino alla visita dello stato ($elapsed\ time$), r è il tempo di

[55] W.M.P. van der Aalst, op. cit, pp. 251-255.

flusso rimanente, s è il tempo di soggiorno nello stato. Lo stato $[a]$ è etichettato con l'annotazione $(t = 12, e = 0, r = 42, s = 7)$ perché è stato visitato dal caso in esame subito dopo l'avvenimento del primo evento a^{12}_{start}; $t = 12$ perché l'evento a^{12}_{start} occorre all'istante 12, $e = 12 - 12 = 0$ perché non è trascorso tempo dall'esecuzione del primo evento, $r = 54 - 12 = 42$ è il tempo rimanente al completamento del caso, $s = 19 - 12 = 7$ perché l'evento successivo accadrà 7 unità di tempo più tardi. Lo stesso ragionamento si adotta per tutte le altre marcature sul modello. Come già anticipato, la Figura 3.7 mostra le annotazioni relative ai primi due casi del log di tabella 3.1 (etichetta nera per il primo caso, etichetta grigia per il secondo caso). Per esempio, lo stato $[p_3, p_4]$ è visitato una volta da entrambi i casi, le cui etichette sono $(t = 33, e = 21, r = 21, s = 2)$ e $(t = 38, e = 21, r = 35, s = 12)$. Lo stato iniziale $[start]$ non presenta annotazioni in quanto non avviene alcun evento visitando tale stato. Lo stato finale $[p_5]$ non ha tempo di soggiorno s perché non vi sono eventi successivi ad esso.

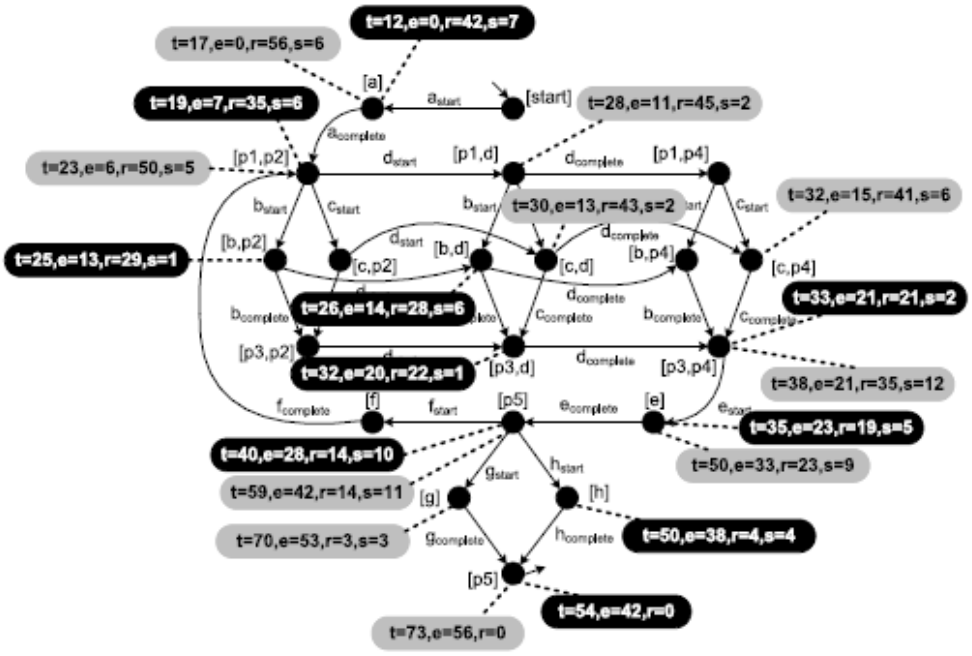

Figura 3.7 Il sistema di transizione con il quale si generano le previsioni dei primi due casi, espresse con le annotazioni t, e, r, s.

Come detto, il log di Tabella 3.1 contiene solo un frammento dell'intero log di eventi; ovviamente, tutti gli altri casi del log possono essere replicati secondo le modalità precedentemente descritte, in modo da raccogliere molte più annotazioni. Ad esempio, il terzo caso visita lo stato $[p_3, p_4]$ due volte: dopo l'evento $d^{40}_{complete}$ e in seguito all'evento $d^{67}_{complete}$. Le corrispondenti annotazioni sono $(t = 40, e = 15, r = 58, s = 5)$ e $(t = 67, e = 42, r = 31, s = 13)$. Se si considerasse un log di notevoli dimensioni, potrebbero esserci per ciascuno stato centinaia o persino migliaia di annotazioni. Per ogni stato x si può definire un vettore riga $Q^{rimanente}_x$ contenente i tempi di flusso rimanenti. Per lo stato $[p_3, p_4]$ tale vettore è $Q^{rimanente}_{[p_3,p_4]} = [21,35,58,31, \dots]$: il primo caso visita lo stato $[p_3, p_4]$ una volta (21 unità di tempo prima del completamento del caso), il secondo caso visita lo stato $[p_3, p_4]$ anch'esso una volta

(35 unità di tempo prima del completamento del caso), il terzo caso visita lo stato $[p_3, p_4]$ due volte (58 e 31 unità di tempo prima del completamento del caso), ecc. Simili vettori esistono anche per le annotazioni e ($Q_{[p_3,p_4]}^{elapsed} = [21,21,15,42, \dots]$) ed s ($Q_{[p_3,p_4]}^{soggiorno} = [2,12,5,13, \dots]$). Con tali vettori è possibile calcolare diverse statistiche. Ad esempio, il tempo di flusso medio rimanente nello stato $[p_3, p_4]$ è:

$$\sum_{q \epsilon Q} \frac{Q(q) \times q}{|Q|} \; con \; Q = Q_{[p_3,p_4]}^{rimanente}$$

Si possono calcolare anche la deviazione standard, i punti di massimo e di minimo, ecc. In seguito, selezionando un campione di dati e utilizzando un software statistico, si può decidere di cercare una distribuzione statistica che ben si adatti ai valori dei tempi di flusso rimanenti. Inoltre, il sistema di transizione può essere usato per la previsione del tempo di flusso rimanente di un caso in esecuzione. La Figura 3.8 illustra il sistema di transizione contenente le annotazioni relative allo stato $[p_3, p_4]$; in aggiunta, la figura evidenzia il cammino relativo ad una traccia parziale del caso in questione. La traccia parziale del caso è: $\left(a_{start}^{512}, a_{complete}^{518}, d_{start}^{525}, d_{complete}^{526}, b_{start}^{532}, b_{complete}^{533}\right)$. Si vuole prevedere il tempo di flusso rimanente all'istante 533. Una buona variabile predittiva è senz'altro la media dei tempi di flusso rimanenti riferita ai casi precedenti e misurata nello stesso istante considerato; come mostrato in Figura 3.8, tale valore è pari a 42.56. Di conseguenza, il completamento del caso è atteso intorno all'istante 575.76. Tale esempio dimostra che per qualsiasi caso, in qualsiasi istante nel tempo, si può prevedere il tempo di flusso rimanente.

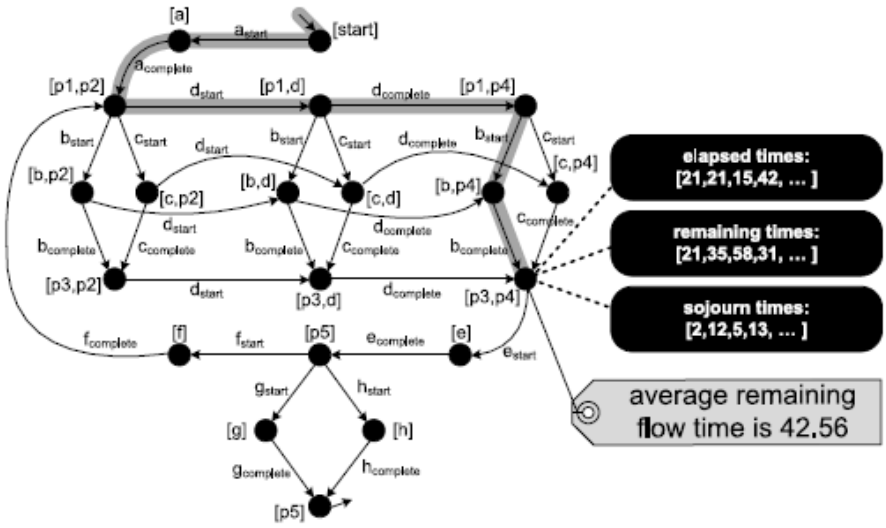

Figura 3.8 Sistema di transizione contenente le annotazioni per lo stato $[p_3, p_4]$ *e il tempo medio di flusso rimanente per la traccia parziale* $\left(a_{start}^{512}, a_{complete}^{518}, d_{start}^{525}, d_{complete}^{526}, b_{start}^{532}, b_{complete}^{533}\right)$.

Oltre a prevedere un singolo valore, la previsione può anche considerare un intervallo di fiducia (es. con un livello di fiducia del 90%, il tempo di flusso rimanente sarà compreso fra 40 e 45 giorni). L'approccio basato sul sistema di transizione con annesse le annotazioni non si limita a prevedere il tempo di flusso; in modo del tutto simile, si possono generare previsioni in merito al tempo di soggiorno. Inoltre, con lo

stesso approccio si possono costruire previsioni non relative al tempo. Riprendendo l'esempio della richiesta di risarcimento, si supponga che si voglia conoscere se una richiesta sarà accolta (avviene l'attività g) o rigettata (avviene l'attività h). Per generare tali previsioni, è necessario annotare per ogni stato gli esiti noti dei casi già completati (post mortem); per esempio, $Q_{[p_3,p_4]}^{accolta} = [0,1,1,1,\dots]$. Per lo stato $[p_3, p_4]$, si aggiunge uno 0 al vettore per ogni richiesta rigettata, mentre si aggiunge un 1 per ogni richiesta accolta. L'esempio mostra che è possibile generare un'ampia varietà di previsioni se si sceglie il sistema di transizione adatto. È importante sottolineare che le informazioni relative al processo sono prese in considerazione, ossia la previsione si fonda sullo stato del caso in esecuzione invece che su attributi statici. La terza e ultima attività del supporto operativo riguarda le raccomandazioni[56]. Come illustrato in Figura 3.9, l'impostazione è analoga alla previsione, ossia all'invio di una traccia parziale al sistema di supporto operativo segue una risposta. Tale risposta, però, non è una previsione ma bensì un suggerimento su quali azioni adottare in seguito, in modo tale da migliorare il processo in esame.

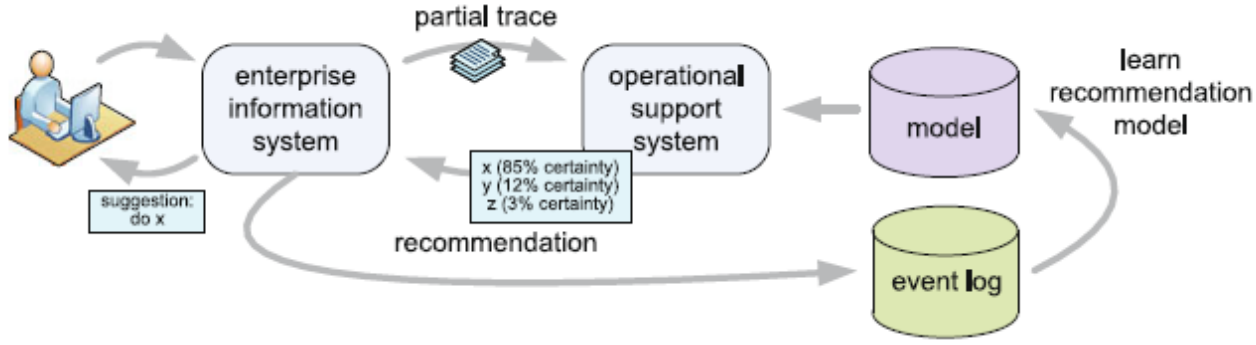

Figura 3.9 Un modello basato su dati storici è lo strumento utilizzato per procurare consigli e suggerimenti circa i casi in esecuzione. Tali suggerimenti non sono costrittivi.

Per fornire suggerimenti, si costruisce un modello sulla base di dati di tipo post mortem; inoltre, è necessario che il sistema di supporto operativo sia a conoscenza dello spazio decisionale, ossia quali siano le possibili azioni fra le quali scegliere. Tali azioni sono ordinate in base ai consigli espressi dal modello. Per esempio, in Figura 3.9 il sistema raccomanda di eseguire l'azione x con un livello di fiducia dell'85%. Le altre due azioni sono meno caldeggiate: le azioni y e z sono consigliate con livelli di fiducia rispettivamente pari al 12% e al 3%. Nella maggior parte dei casi è impossibile garantire che un suggerimento rappresenti la soluzione ottima; la scelta migliore da prendere per il passo successivo può dipendere dal futuro accadimento di eventi ignoti ed esterni al processo. Ad esempio, considerando la Figura 3.9, potrebbero verificarsi eventi in seguito ai quali l'attività z diventi la migliore scelta. Una raccomandazione è sempre data in riferimento a uno specifico obiettivo, come ad esempio minimizzare il tempo di flusso rimanente, minimizzare il costo totale, massimizzare la percentuale di casi completati entro un certo intervallo di tempo, ecc. Le raccomandazioni non sono altro che affermazioni riguardo un insieme di possibili azioni, denotato con il termine spazio decisionale, come già anticipato. Lo spazio

[56] W.M.P. van der Aalst, op. cit, pp. 256-257.

decisionale può essere composto da un insieme di attività, ad esempio $[f, g, h]$; ciò significa che, in riferimento allo stato considerato, le attività f, g e h sono possibili candidate ad essere selezionate come azioni suggerite. Sarà compito del sistema di supporto operativo dare risposta alla domanda "Qual è il miglior candidato considerando l'obiettivo selezionato?". Le raccomandazioni non si limitano al flusso di controllo, possono riferirsi anche ad altre prospettive, come l'allocazione delle risorse. Lo spazio decisionale di un caso in esecuzione può essere parte del messaggio che il sistema informativo aziendale invia al sistema di supporto operativo; in alternativa, il modello che fornisce i suggerimenti dovrebbe essere in grado di derivare lo spazio decisionale sulla base della traccia parziale analizzata. Il suggerimento di un'azione volta a raggiungere un obiettivo è strettamente legato alla previsione del corrispondente indicatore di performance. A seconda della tecnica di previsione utilizzata, la raccomandazione può includere anche informazioni circa la sua affidabilità e la sua qualità, come il livello di fiducia con cui si afferma che una determinata scelta sia ottima rispetto all'obiettivo prefissato. Per esempio, in Figura 3.9 la raccomandazione presenta un livello di fiducia per ognuna delle tre possibili azioni. L'interpretazione di tali valori dipende dal metodo di previsione utilizzato. Con l'uso della simulazione, il livello di fiducia dell'85% riguardo l'azione x di Figura 3.9 significa che, nell'85% degli esperimenti simulati, l'azione x risulta essere la più efficace nella riduzione del tempo di flusso rimanente.

Conclusioni

A valle di quanto discusso nel presente elaborato, si può affermare con ancora più certezza e convinzione che il process mining rappresenti il collegamento mancante fra il data mining e il tradizionale BPM basato sui modelli. Il process mining è un importante strumento a disposizione delle organizzazioni moderne per la gestione di processi operativi complessi. Il mondo digitale e il mondo materiale si amalgamano in un unico universo, dove gli eventi che si verificano sono registrati e conservati nel tempo e i processi sono guidati e controllati sulla base di dati. È stato illustrato il compito più impegnativo del process mining, ossia il process discovery, con annesse le descrizioni delle sue tecniche; tuttavia, il process mining non si limita al process discovery, ma si espande su diverse direzioni. Tali espansioni sono accomunate dallo stretto legame che si instaura fra il modello di processo e il log di eventi, il quale permette lo sviluppo di nuove forme di analisi. Si è discusso del conformance checking, in particolare di come condurre un'indagine finalizzata a stabilire in maniera sia qualitativa che quantitativa l'allineamento fra il modello di processo e il log di eventi in esame, ed è stata declinata la nuova tematica relativa al supporto operativo e alla conduzione di analisi online; in tale maniera, è possibile individuare le discrepanze fra il comportamento osservato nel log e quello espresso dal modello di processo, in modo da acquisire informazioni e consapevolezza riguardo i correttivi necessari da apportare al processo, allo scopo di migliorarlo in modo apprezzabile. Il process mining è utile anche ad affrontare il tema della separazione fra business e Information Tecnnology (IT). Le persone appartenenti al campo dell'IT tendono ad avere una mentalità orientata verso la tecnologia, non curandosi dei processi operativi e del supporto che essi richiedono. Al contrario, le persone operanti in ambito di BPM sono inclini a concentrarsi sulla parte legata al business e, tipicamente, non mostrano interesse verso i progressi tecnologici e la funzionalità dei sistemi informativi. La natura empirica del process mining avvicina i due gruppi di persone e crea un terreno comune comprendente il miglioramento dei processi operativi e lo sviluppo dei sistemi informativi. Sebbene siano attualmente disponibili varie tecniche di process mining, come ampiamente illustrato, sono diverse le sfide che il futuro presenta per migliorare ulteriormente la loro applicabilità. Il professor van der Aalst identifica nel process discovery[57] la sfida più importante e più visibile in merito al process mining; come spiegato in precedenza, è tutt'altro che banale costruire un modello di processo basato su log di eventi incompleti e affetti da disturbi. Sfortunatamente, sono ancora molti i ricercatori e i progettisti che considerano i log come se fossero completi e privi di disturbi. Sebbene l'heuristic mining e il genetic process mining siano tecniche maggiormente insensibili in tal senso, è possibile progredire verso la costruzione di modelli del tipo 80/20, ossia modelli semplici e abili a spiegare la quasi totalità dei comportamenti più frequenti che i processi manifestano. I nuovi approcci al process mining dovrebbero

[57] W.M.P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlino, 2011, p. 339.

riconsiderare l'uso del bias di rappresentazione. Quasi tutti gli approcci esistenti adoperano una notazione che si fonda sui grafi, con i quale si può incorrere nella rappresentazione di modelli di processo poco sensati. Lo spazio di ricerca di una tecnica che si serve di un simile bias di rappresentazione è eccessivamente vasto; per esempio, l'algoritmo α può scoprire reti workflow che non rispettano la condizione di soundness, mentre l'heuristic mining e il genetic process mining possono arrivare alla scoperta di C-nets al cui interno vi sono punti morti. Per tale ragione, il bias di rappresentazione relativo alle tecniche di process discovery necessita di essere affinato, in maniera da consentire la sola rappresentazione di modelli sensibili. Chiaramente, un problema così gravoso richiede nuovi approcci e nuove rappresentazioni per essere risolto. Un altro tema sul quale concentrare gli studi di ricerca è il concept drift, ossia il cambiamento dei processi nel mentre è in corso la loro osservazione; è interessante rilevare quando un processo varia e visualizzare le variazioni stesse. Tuttavia, gli approcci attuali non considerano tali cambiamenti. L'efficacia del process mining dipende fortemente dall'abilità nell'estrazione di log di eventi adatti ai fini dell'analisi. Sfortunatamente, in alcuni sistemi informativi gli eventi sono considerati come prodotti secondari per il debugging oppure sono disseminati su svariate tabelle. Altri sistemi dimenticano gli eventi, ossia perdono dati in memoria; ad esempio, tale circostanza si può verificare sovrascrivendo su eventi datati gli eventi attuali. Con il termine business process provenance si è enfatizzata l'importanza della registrazione dei dati e dell'impossibilità di distorcerli. Pertanto, gli eventi devono essere trattati come elementi di prima classe invece che di rilevanza secondaria. Un'altra sfida che si pone davanti riguarda la costruzione di modelli con livelli di qualità e comprensibilità paragonabili ad una mappa geografica; in tale ottica, il ruolo della cartografia è fondamentale. Infine, è auspicabile che in futuro gli strumenti utili al supporto operativo siano implementati nei sistemi informativi aziendali, allo scopo di promuovere e sviluppare ulteriormente il process mining in modalità online, in modo da renderne concreti i benefici discussi nel presente elaborato.

Bibliografia

1. D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*, MIT Press, Cambridge, MA, 2001.
2. W.M.P. van der Aalst*, Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer-Verlag, Berlino, 2011.
3. W.M.P. van der Aalst, A.K. Alves de Medeiros, e A.J.M.M. Weijters, *Genetic Process Mining*, Department of Technology Management, Eindhoven University of Technology, 2006.
4. A.J.M.M. Weijters, W.M.P. van der Aalst, e A.K. Alves de Medeiros, *Process Mining with the HeuristicsMiner Algorithm*, Department of Technology Management, Eindhoven University of Technology, 2006.
5. W. M. P. van der Aalst, V. Rubin, H. M.W. Verbeek, B. F. van Dongen, E. Kindler e C. W. Günther, *Process mining: a two-step approach to balance between underfitting and overfitting*, Springer-Verlag, Berlino, 2008.