

POLITECNICO DI MILANO

Scuola di Ingegneria dell'Informazione

Corso di Laurea Specialistica in
Ingegneria Informatica



Metodi e tecniche per la gestione di processi personali
e l'integrazione con i social network

Relatore: Prof. Marco BRAMBILLA

Tesi di Laurea di:

Carlo STRACCIALINI
Matr. 739458

Anno Accademico 2010 - 2011

POLITECNICO DI MILANO

Scuola di Ingegneria dell'Informazione

Master of Science in
Computer Engineering



Methods and techniques for personal processes management
and integration with social network

Supervisor: Prof. Marco BRAMBILLA

Master Graduation Thesis:

Carlo STRACCIALINI
Student Id. number: 739458

Academic Year 2010 - 2011

...per essere speciali basta essere se stessi...

*...a chi c'è sempre stato
e sempre ci sarà...*

Sommario

Al giorno d'oggi ci troviamo in un mondo dove la rincorsa all'automazione avanza velocemente, nell'ambiente lavorativo si fa sempre più ricorso a strumenti per la gestione dei processi aziendali in modo da velocizzare e automatizzare le procedure più noiose e ripetitive lasciando all'individuo solo pochi compiti. Al contrario la vita privata procede ad una velocità più ridotta e la quasi totalità degli individui gestisce ancora le proprie attività e l'organizzazione dei propri eventi privati senza sfruttare i mezzi che la tecnologia ha messo a disposizione.

In questo lavoro tratteremo metodi e tecniche che permettano a tutti quanti di accorciare le distanze dal mondo business e ottenere i benefici che l'introduzione dei processi nella vita privata può portare: se poniamo la giusta attenzione ci accorgiamo, infatti, che le problematiche che affrontiamo nella quotidianità spesso si possono ricondurre a dei processi; pensiamo ad esempio all'organizzazione di un party o alla procedura necessaria per l'acquisto di una nuova automobile.

Inoltre ognuno di noi ha il suo profilo pubblicato su qualche social network e ne conosce gli enormi vantaggi che queste nuove piattaforme portano con loro: per questo motivo abbiamo deciso anche noi di inglobare le loro funzionalità più interessanti nella nostra soluzione: in modo da dare ai nostri utenti la possibilità di gestire le attività in collaborazione con i propri amici e conoscenti.

Abstract

Today we find ourselves in a world where the pursuit of automation advances rapidly, the workplace becomes increasingly use tools for managing business processes to speed and automate the most tedious and repetitive procedures to leaving 'only a few individual tasks. In contrast, the private life proceeds at a lower speed and nearly all of the individuals still runs its activities and organization of their private events without using the resources that technology has made available.

In this work we will discuss methods and techniques that allows everybody to shorten the distance from the business world and gain the benefits that the introduction of processes may result in private life: if we pay the right attention, we find, in fact, that the problems we face in everyday life can often be traced back to processes, consider the example of a party organization or procedure required to purchase a new car.

In addition, each of us has a profile published on some social network and knows the enormous benefits that these new platforms bring with them: for this reason we have decided to incorporate their most interesting features in our solution: to give our users the ability to manage activities in collaboration with their friends and relatives.

Indice generale

Sommario.....	6
Abstract.....	7
Indice generale.....	9
Elenco delle figure.....	11
1. Introduzione	13
2. Background	17
2.1. Social task management tools.....	17
2.2. BPM e BPMN.....	29
2.2.1. Business Process Modeling	29
2.2.2. Business Process Modeling Notation	30
2.3. Social BPM.....	32
2.4. Soluzioni enterprise per il Social BPM.....	34
2.4.1. IBM Blueworks Live	34
2.4.2. Intalio Social BPM	37
2.5. Personal Process Management.....	41
3. Approccio	43
3.1. Introduzione	43
3.2. Modello concettuale.....	45
3.2.1. Modello utente.....	46
3.2.2. Modello task	47
3.2.3. Modello processo.....	49
3.2.4. Controllo di ciclicità	52
3.3. Integrazione social	55
3.4. Esempi.....	57
4. Implementazione	60
4.1. Analisi dei requisiti	60
4.1.1. Autenticazione utente	60
4.1.2. Inserimento e completamento task	61
4.1.3. Goal (obiettivo).....	62
4.1.4. Gestione contatti	63
4.1.5. Integrazione social network.....	64
4.2. Mockup interfaccia	65

4.2.1.	Registrazione e autenticazione	66
4.2.2.	Home page utente	67
4.2.3.	Dettagli task	68
4.2.4.	Dettagli goal	68
4.2.5.	Gestione contatti	70
4.3.	Ipotesi semplificative e soluzioni adottate	71
4.3.1.	Programmazione grafica	71
4.3.2.	Copia dei processi avviati	71
4.3.3.	Assegnamento task ad utenti esterni	72
4.4.	Design	72
4.4.1.	Architettura	72
4.4.2.	Struttura database	73
4.4.3.	Diagramma delle classi	78
4.5.	Tecnologie utilizzate e dettagli implementativi	81
4.5.1.	Linguaggi di sviluppo	81
4.5.2.	Dettagli implementativi	82
5.	Esempi d'uso	84
5.1.	Aspetti generali	84
5.2.	Esempio – Gestione processo	86
5.3.	Esempio – Integrazione social network	93
5.4.	Valutazioni utenti	96
6.	Conclusioni	98
	Bibliografia	101
	Ringraziamenti	104

Elenco delle figure

Figura 2.1 – Estensioni del Social BPM.....	33
Figura 2.2 – IBM Blueworks Live.....	35
Figura 2.3 – Ciclo di vita Intalio Social BPM	37
Figura 3.1 – Modello concettuale della soluzione proposta	45
Figura 3.2 – Esempio di task sequenziali	50
Figura 3.3 – Esempio di task paralleli	51
Figura 3.4a – Ciclo: struttura semplice.....	53
Figura 3.4b – Ciclo: struttura complessa	54
Figura 3.4c – Processo senza cicli	54
Figura 3.5 – Esempio: organizzazione partita a calcetto	57
Figura 3.6 – Esempio: progetto scolastico o universitario	58
Figura 4.1 – Esempio struttura goal.....	62
Figura 4.2 – Login e registrazione.....	66
Figura 4.3 – Home page utente.....	67
Figura 4.4 – Dettagli di un task	68
Figura 4.5 – Definizione struttura obiettivo	69
Figura 4.6 – Pagina dei contatti	70
Figura 4.7 – Modello concettuale	73
Figura 4.8 – Modello logico-relazionale	74
Figura 4.9 – Diagramma delle classi	78
Figura 5.1 – Home page utente.....	84
Figura 5.2 – Pagina contatti.....	85
Figura 5.3 – Creazione processo.....	86
Figura 5.4 – Dettagli goal	87
Figura 5.5 – Creazione processo.....	88
Figura 5.6a – Situazione iniziale processo: vista creatore.....	88
Figura 5.6b – Situazione iniziale processo: vista utente.....	89
Figura 5.7 – Primo task completato.....	90
Figura 5.8 – Task e deadline.....	90

Figura 5.9 – In attesa del Task C per completare l'ultimo task.....	91
Figura 5.10a – Obiettivo completato	92
Figura 5.10b – Dettagli obiettivo completato	92
Figura 5.11 – Login con Facebook	93
Figura 5.12 – Invito tramite Facebook	94
Figura 5.13 - Processo social	94
Figura 5.14 – Notifica Task 1	95
Figura 5.15a – Completamento Task 1: sistema.....	95
Figura 5.15b – Completamento Task 1: notifica	96

1. Introduzione

Nell'era del social web, ogni navigatore ha ormai un proprio profilo per farsi conoscere su svariati social network, dai sempre più popolari Facebook e Twitter a piattaforme maggiormente indirizzate ad ambienti lavorativi come LinkedIn, e ciò ha favorito un aumento delle nascite di nuove relazioni tra gli utenti: questo perché seguono passioni comuni, hanno gli stessi hobby, lavorano nello stesso settore anche se a chilometri di distanza e non hanno mai avuto il piacere di conoscersi di persona. Ma è proprio questo favorire e far nascere nuovi rapporti il vero punto di forza dei social network.

Dall'altra parte gli utenti continuano a vivere ed organizzare la propria vita molto spesso non sfruttando le potenzialità offerte da questo nuovo mondo social: attività che coinvolgono più di un individuo sono sempre gestite nella vecchia maniera, con telefonate, appuntamenti e messaggi e spesso si corre il rischio di incomprensioni e ritardi nello stesso gruppo per problemi di comunicazione e coordinamento. Sebbene ciò possa essere accettabile nella gestione della propria vita privata, non è assolutamente pensabile il verificarsi di queste situazioni in ambienti lavorativi dove sono richieste professionalità e serietà e, per questo motivo, nelle aziende e nei servizi si è già fatto ricorso a strumenti informatici adatti alla gestione automatica dei processi di business in modo tale da coordinare le attività di dipendenti, manager e team ed automatizzare, ove possibile, alcune procedure.

Nella vita privata, invece, l'utilizzo di questi strumenti integrati con soluzioni social è pressoché nullo. Molti individui fanno ancora ricorso ai vecchi post-it per appuntare le cose da fare durante la giornata oppure, nel caso volessero organizzare qualche tipo di evento come una festa o una cena insieme ad altre persone, devono mettersi d'accordo su chi deve fare cosa e molto spesso questo è causa di incomprensioni. Questo problema, nell'era dell'Information Technology, può essere risolto se trasferiamo alcune idee del mondo business come la gestione dei processi e le applichiamo nella vita di tutti i giorni: ecco quindi come possiamo risolvere il problema di gestire le nostre attività, singole oppure insieme ad altri individui, nella vita reale, scomponendo

quelle più complesse e pensandole come dei processi da portare a termine con il coinvolgimento di tutti gli individui assegnando loro determinati compiti.

La soluzione proposta in questa tesi si propone di seguire questa idea: affiancare ai normali compiti di un gestore di tasklist, la capacità di gestire strutture complesse di task che possiamo pensare come dei processi di tipo aziendale e sfruttare le potenzialità offerte dai social network per coinvolgere il maggior numero di utenti possibili nelle nostre attività. L'approccio pensato è quello di sviluppare un'applicazione su piattaforma web che permetta la definizione di processi in maniera semplice e rapida, senza troppe impostazioni che possano rendere difficile l'apprendimento e l'utilizzo per l'utente non esperto di sistemi informatici.

Il vero punto di forza della nostra soluzione, infatti, vuole essere la semplicità: nel nostro modello di processo sono stati eliminati alcuni costrutti normalmente presenti in prodotti enterprise destinati alle aziende ma poco adatti per gli scopi di un utente medio e la gestione delle attività quotidiane; per quanto complicata possa essere la gestione contemporanea di task finalizzati verso un unico obiettivo non saranno mai raggiunti i livelli di complessità di processi business. Inoltre abbiamo cercato di integrare i metodi e le tecniche più intuitive per la fase di modellazione dei processi e l'assegnamento ad utenti come, ad esempio, il drag&drop che permette quindi una sorta di modellazione grafica.

Il secondo punto di forza della nostra idea è quello di sfruttare gli strumenti e la rete di contatti e amici che ci può offrire l'integrazione dei social network nella nostra applicazione. Basti pensare ai vantaggi che arrivano da un'autenticazione effettuata per mezzo di un social network come Facebook: innanzitutto l'utente non necessita di una nuova coppia di credenziali per autenticarsi nel sistema, porta con sé tutta la sua rete di contatti ai quali può chiedere di registrarsi nel sistema e poi assegnare attività e task di processi una volta iscritti ed, infine, può sfruttare il sistema di notifiche in dotazione al social network per comunicare eventuali assegnamenti o completamenti.

Dopo questo breve capitolo introduttivo, la tesi si svilupperà nei seguenti capitoli:

- **Capitolo 2 – Background:** nella prima sezione analizzeremo le soluzioni presenti nel web per la gestione di semplici tasklist personali, cercando di evidenziare gli aspetti positivi e quelli negativi; successivamente presenteremo gli aspetti caratteristici del BPM e del SocialBPM ed approfondiremo l'analisi di alcuni strumenti informatici adottati in ambienti enterprise per il SocialBPM sviluppati dalle più importanti software house del settore.
- **Capitolo 3 – Approccio:** in questo capitolo approfondiremo il nostro approccio risolutivo al problema affrontato; cercheremo di formalizzare attraverso un modello matematico i concetti che compaiono nella nostra soluzione ed in particolare ci soffermeremo sull'aspetto principale del nostro lavoro, cioè il concetto di processo, illustrando alcune caratteristiche che deve possedere come ad esempio il fatto di non presentare cicli per garantire il corretto funzionamento dell'applicazione finale. Concluderemo quindi soffermandoci su alcuni aspetti dell'integrazione social e presentando alcuni esempi di modello di processo.
- **Capitolo 4 – Implementazione:** tratteremo l'implementazione reale della nostra applicazione seguendo il tipico approccio dettato dall'ingegneria del software; analizzeremo i requisiti della nostra soluzione e proporremo alcune idee per l'interfaccia utente; in seguito passeremo a discutere della fase di modellazione delle classi e mappatura dei concetti e quindi illustreremo i dettagli implementativi del nostro lavoro e le tecnologie utilizzate per lo sviluppo.
- **Capitolo 5 –** In questo capitolo presenteremo l'applicazione implementata con alcuni screenshot e mostreremo due esempi con i quali illustreremo le caratteristiche della nostra soluzione: nel primo metteremo in luce gli aspetti legati alla definizione e gestione dei processi, nel secondo gli aspetti innovativi legati all'integrazione con il social network Facebook; infine concluderemo con una breve valutazione globale dell'applicazione sviluppata raccogliendo alcuni feedback di diversi utenti.

- Capitolo 6 – Conclusioni: ribadiremo gli obiettivi del nostro lavoro e verificheremo se sono stati raggiunti, fornendo anche una valutazione critica, che segue anche i commenti di chi ha utilizzato realmente l'applicativo, su quali aspetti è possibile migliorare e quali invece si sono rivelati dei veri punti di forza; esporremo infine dei possibili sviluppi futuri di questo lavoro proponendo alcune idee interessanti.

2. Background

2.1. Social task management tools

Nel vasto mondo del Web sono presenti molti strumenti che supportano l'utente finale nella gestione dei propri task personali e nell'organizzazione delle attività quotidiane; essi, però, mancano completamente sotto l'aspetto della gestione di processi più articolati o di task tra loro logicamente collegati. Ciononostante alcuni di loro offrono alcune caratteristiche importanti che in seguito andremo ad esaminare in modo più approfondito tralasciando quelle soluzioni che non offrono idee o spunti interessanti ma svolgono solamente il semplice compito di sostituzione di post-it o blocco-note. Nella successiva Tabella 2.1 riportiamo l'elenco completo di tutte le soluzioni online analizzate con indicato l'indirizzo web attraverso il quale possiamo raggiungerle.

Tool	Url
Remember The Milk	http://www.rememberthemilk.com
Online Task List	http://www.onlinetasklist.com/
Hi Task	http://hitask.com/
Todoist	http://todoist.com/
Toodledo	http://www.toodledo.com/
Tadalist	http://tadalist.com/
Voo2do	http://voo2do.com/
Astrid	http://astrid.com/
Cozi	http://www.cozi.com/
Blablalist	http://blablalist.com/
CCtodo	http://cctodo.com/
Tasktoy	http://www.tasktoy.com/
GTDagenda	http://www.gtdagenda.com/

Manymoon	https://manymoon.com/
Producteev	http://www.producteev.com/
Workhack	http://workhack.com/
Webtodo	http://webtodo.wndmll.com/
Theonlineceo	http://roughunderbelly.com/user/login
Nozbe	http://www.nozbe.com/
Tedium	http://www.mcqn.com/tedium/account/login
Checkvist	http://checkvist.com/
Hiveminder	http://hiveminder.com/splash/
Stayuseful	http://stayuseful.com/
Nutshell	http://www.gonutshell.com/

Tabella 2.1 – Elenco tools online

Per ogni singola web-application, abbiamo identificato le funzionalità offerte e le abbiamo inserite in una tabella in modo da offrire una scheda riepilogativa e comparativa tra tutte le soluzioni prese in considerazione. Qui di seguito descriviamo le feature rilevate e ne illustriamo le caratteristiche, chiarendo eventuali dubbi legati al loro significato.

- Integrazione con i social network: nei casi più semplici equivale all’associazione dei dati di login e quindi evita un ulteriore registrazione ma può diventare più complessa se con essa si intende la possibilità di pubblicare post sulle bacheche o wall degli utenti o di interagire con la sua rete di amici e contatti.
- Contest/Categorie/Tag: corrispondono ad etichette associate ai task per raggrupparli secondo alcuni criteri; ad esempio un tag “Home” può riferirsi a task in qualche modo riconducibili alle cose da fare in casa. Il loro inserimento può avvenire attraverso la selezione in un campo opportuno oppure marcando il titolo del task particolari espressioni, chiamate appunto tag.

2. BACKGROUND

- Gestione località: è un'integrazione con Google Maps che permette l'indicazione del luogo dove vengono eseguiti i task (se è stato impostato la località tra le sue proprietà) su di una mappa geografica.
- Sincronizzazione esterna: permette di sincronizzare il proprio device mobile con la propria tasklist così l'utente può avere sempre sotto controllo le sue attività quotidiane.
- Gestione progetti: è una funzionalità più adatta ad imprese e team lavorativi che a semplici utenti finali (ma può rivelarsi utile anche a quest'ultimo ad esempio con un fantomatico progetto di festa di compleanno); può corrispondere ad una semplice catalogazione di task nei casi più semplici, fino ad una vera e propria gestione di un progetto con una lista di task, la lista dei membri del team, eventuali file caricati dai partecipanti al progetto ed anche un bacheca dove parlare di idee, discutere dubbi e problemi e proporre eventuali soluzioni.
- Condivisione pubblica task: permette di generare una pagina web che contiene la propria tasklist con accesso pubblico, cioè un qualsiasi navigatore di Internet può accedere alla pagina e visualizzare i task ma non può compiere operazioni di modifica.
- Modifica permessi utenti: qualora si abbia la possibilità di lavorare in un team composto da più utenti per il completamento di un qualche progetto, l'amministratore o il supervisore può modificare i permessi ai vari collaboratori, garantendo o negando alcune operazioni: alcuni esempi di permesso possono essere visualizzazione, modifica e commento.
- Drag&Drop per operazioni comuni: in alcuni casi l'utente può associare task a categorie, impostare date di scadenze, assegnare task a membri del team semplicemente trascinando il task sulla rispettiva opzione facilitando notevolmente l'uso dello strumento a disposizione.
- Time tracking: una funzione con la quale l'utente può cronometrare l'effettivo tempo di completamento di un task; nella maggior parte delle situazioni i dati raccolti saranno tenuti in considerazione nella generazione di statistiche e report.

- **Stampa booklet:** permette la creazione di semplici booklet da stampare utilizzando alcuni template predefiniti e fornendo precise istruzioni per il taglio e la piega delle stampe ottenute in modo da ricavare comodi libricini trasportabili.
- **Checklist periodiche:** sono liste associate a determinati periodi temporali (settimane, mesi) utili per la ripetizioni di task quotidiani; con esse è possibile organizzare facilmente lezioni universitarie, allenamenti, ecc; riunisce le funzioni di una tasklist con quelle di un agenda.

Esaminiamo ora ad uno ad uno i casi di studio evidenziandone alcune caratteristiche peculiari e cercando di mettere in risalto i pro e i contro di ogni sistema.

Remember The Milk: risulta essere uno dei sistemi più semplici ed immediati ma manca di alcune feature avanzate che lo avrebbero reso adatto anche in contesti aziendali: si nota la mancanza di una gestione di progetti. Molto ben implementata risulta la gestione dei contatti e le funzioni ad essa associate come la condivisione di task o liste; per la parte di notifica ci si può affidare ai più comuni software di messaggistica online oltre alla semplice mail ed è anche possibile sincronizzare i propri device (a pagamento).

Pro	Contro
<ul style="list-style-type: none"> • Semplice ed immediato • Gestione contatti • Notifiche • Località • Smart tag (@data, @location, ecc.) 	<ul style="list-style-type: none"> • Manca una gestione dei progetti • Pochi formati di import/export

Online Task List: orientato a contesti aziendali grazie all'ottimo supporto a progetti e team, include inoltre la possibilità di caricare file associati al progetto ed editare a proprio piacimento stati e priorità; fornisce anche strumenti per visualizzare statistiche

e permette agli utenti di un team di aggiungere commenti ai task. La grafica spartana e il tipo di funzionalità offerte lo rende, però, inadatto ad un pubblico che ricerca semplicità e immediatezza in questi software.

Pro	Contro
<ul style="list-style-type: none"> • Adatto ad aziende con progetti e team • Ricco di funzionalità • Upload file • Statistiche • Diversi ruoli all'interno dei team 	<ul style="list-style-type: none"> • Poco user-friendly • Solo supporto CSV • No inserimento veloce • Differente finestra per singoli task e task ripetitivi

Hi Task: Si nota dall'interfaccia gradevole e pulita la sua ricca dotazione di feature; come prima cosa notiamo che include anche una calendario per utilizzarlo anche come agenda infatti oltre a task possiamo inserire eventi e note; è possibile inserire più task contemporaneamente caricando un file CSV opportunamente formattato e uploadare file. Offre numerosi formati di esportazione.

Pro	Contro
<ul style="list-style-type: none"> • Ottima interfaccia • Ricco di funzionalità • Time tracking • Gerarchie di task • Drag&Drop 	<ul style="list-style-type: none"> • No località • Integrazione con social network

Todoist: presenta funzionalità minime ma alcune interessanti come la presenza di progetti e sub-task; non esiste un indice di priorità ma un'indicazione di ciò è data dall'ordinamento dei task che può essere editato tramite drag&drop. Infine sono utilizzati tag e date per creare viste personali dei task da completare.

Pro	Contro
<ul style="list-style-type: none"> • Di immediato utilizzo • Tag per creare viste personali 	<ul style="list-style-type: none"> • Manca supporto team • Nessun controllo su date tra task e sub-task • No import/export

Toodledo: rispetto ad altri introduce il concetto di “Goal” che corrisponde ad un obiettivo da portare a termine attraverso il completamento di determinati task; nella versione a pagamento esiste anche un’utility che tiene traccia dei progressi e uno scheduler che aiuta a gestire al meglio il tempo a disposizione. Per il resto l’applicativo, nella sua versione free, offre solo le classiche funzionalità di base per una semplice gestione di una tasklist.

Pro	Contro
<ul style="list-style-type: none"> • Interfaccia semplice • Concetto di “Goal” 	<ul style="list-style-type: none"> • Molte feature disponibili solo a pagamento

Tadalist: basato su un insieme di liste personalizzabili, non offre altro che la creazione di task senza però poterne specificare alcuna caratteristica come data, descrizione, ecc. e il loro ordinamento con il sistema del Drag&Drop.

Pro	Contro
	<ul style="list-style-type: none"> • Non si possono editare proprietà task • Nessuna feature avanzata

Voo2do: è un sito dalla grafica abbastanza spartana ma che svolge bene le operazioni per il quale è stato progettato, includendo alcune caratteristiche più avanzate come la gestione di progetti e collaboratori; purtroppo non offre un metodo per un inserimento rapido dei task obbligando l’utente a completare ogni campo del task (e non sono

pochi proprio perché offre le funzionalità sopra elencate) e non è stato implementato un sistema di notifica o remainder.

Pro	Contro
<ul style="list-style-type: none"> • Gestione semplice • Progetti e collaboratori • Time tracking per le statistiche 	<ul style="list-style-type: none"> • No inserimento rapido • Solo XML per import/export • Nessun sistema di notifica • Interfaccia migliorabile

Astrid: piattaforma di lavoro molto spinta verso l'ambito social, l'interfaccia ricorda infatti alcuni popolari social network, ma non è solo questo che lo rende molto simile: infatti è possibile avere una lista di amici, condividere con essi task, commentarli e seguire lo sviluppo dei task degli amici. Ma proprio questa somiglianza lo allontana ad un possibile utilizzo in ambito aziendale dato che è completamente assente una sezione dedicata a progetti e caricamento di file ad esso associati così come manca un sistema di notifica diverso dalla classica mail.

Pro	Contro
<ul style="list-style-type: none"> • Orientato al social • Ottima integrazione con Twitter e Facebook • Ottima gestione dei task 	<ul style="list-style-type: none"> • Progetti • Sezione "Amici" migliorabile • Sistemi di notifica • Upload file • Esportazione assente

Cozi: è destinato ad un uso familiare infatti si crea un unico account associato a tutti i membri della famiglia, in questo modo la condivisione dei task è immediata; si nota la destinazione d'uso anche dalle varie possibilità offerte oltre la semplice tasklist come la lista della spesa, il calendario ed altre ancora. Questo applicativo, però, ideato in questo modo non può rispondere alle esigenze di molte altre tipologie di utente perché ogni sua caratteristica è stata pensata in funzione della sua propensione all'uso familiare.

Pro	Contro
<ul style="list-style-type: none"> • Utilizzo familiare • Calendario, lista spesa, bacheca oltre tasklist 	<ul style="list-style-type: none"> • Non adatto ad alcun utente se utilizzato al di fuori dell'ambito familiare • Niente date ma solo priorità

Blablalist: un sistema molto minimal che permette all'utente di creare più liste e aggiungere task; si può, eventualmente, condividere privatamente la lista con altri utenti tramite mail, i quali possono apportare le loro modifiche ma non eliminarla o condividerla a loro volta con altri, o renderla visibile a chiunque rendendola pubblica tramite pagina web e feed RRS 2.0.

Pro	Contro
<ul style="list-style-type: none"> • Semplice • Minimo • Condivisione privata e pubblica di intere liste 	<ul style="list-style-type: none"> • Nessuna caratteristica avanzata • No condivisione singoli task

CCtodo: anch'esso offre solamente funzionalità di base per l'inserimento dei task, oltretutto non permettendone un inserimento rapido a costringendo l'utente ad inserire tutte le caratteristiche del task stesso; l'unico aspetto di rilievo è l'uso di tag per raggruppare task relativi ad uno stesso contesto. Esistono anche molti porting verso altri tipologie di piattaforma, mobile e desktop, come ad esempio Android, Mac, iPhone, ecc.

Pro	Contro
<ul style="list-style-type: none"> • Multi-piattaforma 	<ul style="list-style-type: none"> • Funzionalità minime

Tasktoy: quest'applicazione è destinata ad un ambito business, l'interfaccia grafica è molto scarna però offre tutte le caratteristiche adatte all'uso aziendale come progetti, feed e note. Dispone di due comodi menù a tendina al fianco di ogni task con i quali si può immediatamente modificare la data e l'etichetta del tag, senza dover entrare nella vista relativa ai dettagli del tag stesso. Manca però di un sistema di notifica e di funzioni per l'import/export delle tasklist oltre alla completa assenza di tutta la gestione dei contatti e amici.

Pro	Contro
<ul style="list-style-type: none"> • Immediato • Rapida modifica data e contest • Feed 	<ul style="list-style-type: none"> • Notifiche • Import/Export • Contatti

GTDagenda: è uno degli esempi più interessanti in quanto a numero di funzionalità presenti, allo stesso tempo ne preclude leggermente la facilità d'uso; è orientato principalmente verso ambienti di tipo business, si adatta facilmente anche ad altri contesti poiché è molto personalizzabile. Tra le caratteristiche innovative che offre possiamo ricordare la creazione di mail distinte per ogni categoria esistente, un mark ad indicare il prossimo task da completare, l'aggiunta dell'attributo stato relativo ai progetti, la gestione di checklist relative a determinati frame temporali e ritorna, come già avevamo visto in un altro caso, il concetto di goal. Unica pecca è la gestione dei contatti e amici solo nella versione a pagamento.

Pro	Contro
<ul style="list-style-type: none"> • Completamente personalizzabile • Next action • Goal e progetti • Mail distinte per categorie 	<ul style="list-style-type: none"> • Contatti e amici solo nella versione a pagamento

2. BACKGROUND

Manymoon: anche quest'applicativo risulta essere molto completo e particolarmente orientato a progetti condivisi tra più utenti, rimane quindi in un ambito aziendale. In particolare la sezione dei progetti è gestita in modo ottimale con diverse sezioni, ognuna delle quali ha una ben specificata funzione: bacheca, task, upload file, goal, eventi, membri, timesheets e report; sono inoltre disponibile alcuni template di progetto e molti altri sono disponibili online con una semplice ricerca. Inoltre una piccola peculiarità vista solo in questo caso riguarda l'assegnazione di uno stesso task a più persone: si può decidere se il task sarà marcato come completo se almeno uno degli assegnatari lo ha completato o se necessariamente tutti sono vincolati nello svolgerlo.

Pro	Contro
<ul style="list-style-type: none">• Project-oriented• Template di progetto• Integrazione con Google e LinkedIn• Controllo dei permessi dei partecipanti al progetto	<ul style="list-style-type: none">• Nulla di particolare

Producteev: basato su dei workspace, ognuno associato ad un unico utente, offre le classiche feature per un sistema di questo genere adatto a qualsiasi contesto e un'interfaccia pulita simile a quella dei social network con i quali ha un elevato tasso di integrazione. Il workspace, in pratica, racchiude tutto l'ambiente di lavoro dell'utente ed è visualizzato, oltre che da se stesso, dai suoi amici (molto simile ad una bacheca condivisa con gli amici).

Pro	Contro
<ul style="list-style-type: none">• Integrazione con i più popolari social network• Basato sui workspace	<ul style="list-style-type: none">• Nulla di particolare

Nel seguito indichiamo altri esempi considerati minori perché offrono solamente caratteristiche di base e ne metteremo in luce alcune feature particolari che non sono apparse nei casi precedenti

- Workhack: l'unica peculiarità sta nella rappresentazione grafica del livello di priorità che varia dimensione e colore per ogni livello disponibile.
- Webtodo: inserimento dei task distinto in base alla scadenza, breve medio e lungo termine; esiste inoltre la possibilità di editare queste categorie e di aggiungerne di nuove.
- Theonlineceo: propone solo la gestione di una tasklist senza offrire alcuna funzionalità aggiuntiva o qualche aspetto di un certo interesse.
- Nozbe: permette la conversione automatica di task formati da sub-task in progetti.
- Tedium: permette la personalizzazione dell'interfaccia grafica impostando quante e quali liste di task visualizzare e come disporle nell'area di lavoro. Genera automaticamente report ad intervalli di tempo predefiniti.
- Checkvist: fornisce supporto per import/export in OPML 2.0

Sono stati analizzati ulteriori tre esempi, Hiveminder, Stayuseful e Nutshell, i quali non offrivano nulla di particolarmente interessante e innovativo rispetto ai casi già esaminati.

Nella Tabella 2.2 abbiamo messo a confronto i vari tools analizzati ed indicato quali caratteristiche offrono all'utilizzatore finale.

2. BACKGROUND

	Remember the milk	Onlinetasklist	Hitask	Todoist	Toodledo	Tadalist	Voo2do	Astrid	Cozi	Blablist	Ctodo	Theonlineeo	Taskboy	Workhach	Webtodo	GTDagenda	Manuymoon	Producteev
Integrazione con social network	Si				Si			Si								Si	Si	Si
Aggiunta task via mail	Si		Si				Si									Si		Si
Inserimento veloce	Si		Si		Si			Si	Si	Si		Si	Si	Si	Si		Si	Si
Contest/Categorie/Tag	Si	Si	Si	Si	Si		Si	Si			Si		Si			Si	Si	Si
Gestione località	Si				Si													
Gestione contatti	Si	Si	Si			Si	Si	Si	Si							Si	Si	
Sviluppo API	Si																	
Export	Si	Si	Si	Si	Si		Si							Si		Si		
Import		Si			Si		Si									Si		Si
Feed	Si		Si			Si				Si			Si	Si				
Ricerca	Si		Si	Si							Si					Si		Si
Notifiche	Si	Si						Si										Si
Sincronizzazione	Si																	Si
Keyboard shortcut	Si	Si		Si				Si										
Gestione progetti		Si	Si	Si			Si						Si			Si	Si	
Editing permessi		Si															Si	
Assegnamento task		Si	Si				Si	Si	Si	Si	Si							Si
Condivisione pubblica		Si	Si			Si	Si	Si		Si			Si	Si				
Statistiche/report		Si	Si				Si	Si										Si
Drag&Drop			Si	Si		Si	Si				Si		Si	Si			Si	
Upload file		Si	Si		Si												Si	
Time tracking			Si		Si		Si											
Inserimento multiplo			Si		Si													
Backup/Restore			Si		Si		Si											
Stampa booklet			Si		Si													
CheckList periodiche																Si		

Tabella 2.2 – Scheda comparativa

Come possiamo osservare alcuni strumenti sono veramente molto completi come numero di feature offerte all'utente come Hitask, Onlinetasklist, Astrid e Remember The Milk ma nessuno di essi un sistema per la gestione di processi personali come quelli che vogliamo gestire noi. Nonostante questo alcune caratteristiche possedute da questi tools come il drag&drop e l'assegnamento dei task si sono rivelate adatte ai nostri scopi e per questo motivo sono state integrate nella nostra soluzione.

2.2. BPM e BPMN

2.2.1. Business Process Modeling

Nell'ingegneria dei sistemi, il Business Process Modeling (BPM) corrisponde all'attività di rappresentare un processo in ambiente enterprise in modo tale da potere essere analizzato ed, eventualmente, migliorato. È un'attività tipicamente svolta da analisti e manager che stanno puntando a migliorare efficienza e qualità di determinati processi e può essere coadiuvata dall'utilizzo di strumenti informatici per le operazioni più ripetitive sebbene questo non sia strettamente necessario.

Nell'ambito di questa attività possiamo definire alcuni termini per descrivere alcuni concetti chiave di questo mondo:

- 1) Il business model è una piattaforma utilizzata per una vasta gamma di descrizioni formali ed informali che può rappresentare aspetti centrali del mondo business, inclusi scopi, offerte, strategie, infrastrutture, strutture organizzative, pratiche di commercio, processi operazionali e linee d'azione.
- 2) Il business process è una collezione di attività o task strutturate o relazionate per produrre uno specifico servizio o prodotto (o servire ad uno specificato obiettivo) per particolari clienti; esistono tre principali tipologie di business process:
 - processi di gestione, che governano le operazioni del sistema; sono incluse, ad esempio, le strategie di gestione;

- processi operazionali, che costituiscono gli aspetti fondamentali del business e il flusso principale di lavoro; tipici processi sono quelli relativi ad acquisti, produzione, marketing e vendite;
- processi di supporto, che cooperano e supportano i processi centrali del business come il supporto tecnico e il reclutamento.

Un business process può essere decomposto in molti processi figli, ognuno dei quali ha i propri attributi, ma contribuiscono tutti al raggiungimento dell'obiettivo del processo padre. L'analisi dei business process in genere include la mappatura dei processi e dei sotto-processi fino al livello di attività. Un business process model è un modello di uno o più business process, e definisce i modi in cui sono effettuate le operazioni per raggiungere gli obiettivi previsti dall'organizzazione. Tale modello rimane un'astrazione e dipende dall'uso previsto del modello stesso. Con esso si può descrivere il flusso di lavoro o l'integrazione tra i business process. Esso può anche essere costruito in molteplici livelli.

2.2.2. Business Process Modeling Notation

La Business Process Management Initiative (BPMI) ha sviluppato la prima specifica dello standard Business Process Modeling Notation (BPMN 1.0) nel maggio del 2004; nel momento in cui scriviamo, siamo in possesso della specifica 2.0 di BPMN, rilasciata agli inizi del 2011, che è anche diventata uno standard dell'Object Management Group (OMG) [10]. L'obiettivo principale degli sforzi per lo sviluppo di BPMN è stato quello di fornire una notazione che risultasse comprensibile da tutte le tipologie di utenti business, dall'analista che crea lo schizzo iniziale dei processi, passando allo sviluppatore tecnico responsabile dell'implementazione tecnologica che eseguirà i processi, arrivando all'utente lavorativo che gestirà e monitorerà quei processi. BPMN sarà anche supportato da un modello interno che darà la possibilità di generare codice eseguibile BPEL4WS. È quindi una sorta di ponte per il gap che intercorre tra il progetto di un business process e la sua realizzazione implementativa. BPMN definisce un Business Process Diagram (BDP) per la creazione modelli grafici delle operazioni del processo lavorativo, basato sulla tecnica dei diagrammi di flusso

orientati. Un Business Process Model è quindi composto da una rete di oggetti grafici i quali rappresentano attività o task, e il flusso di controllo definisce il loro ordine di esecuzione.

Un BDP è formato da un insieme di elementi grafici. Questi elementi facilitano lo sviluppo di semplici diagrammi che, in questo modo, risulteranno molto familiari alla maggior parte degli analisti aziendali. Gli elementi sono stati scelti per essere distinguibili gli uni dagli altri ed utilizzano forme comuni alla maggior parte dei modellisti; per esempio le attività sono rappresentate da rettangoli mentre le decisioni da rombi. In questo modo si è cercato di mostrare come uno dei principi sui quali è stato basato lo sviluppo di BPMN è quello di creare dei semplici meccanismi per la definizione dei modelli dei processi aziendali, mantenendo allo stesso tempo la capacità di gestire in modo ottimale la complessità degli stessi processi. L'approccio utilizzato per gestire questi due requisiti che sono tra loro in conflitto è stato di organizzare gli aspetti grafici della notazione in categorie ben specificate. Questo piccolo insieme di categorie fornisce al lettore di un BDP tutte le nozioni necessarie affinché possa facilmente riconoscere il tipo degli elementi e comprendere quindi il diagramma. Naturalmente, variazioni e nuove informazioni possono essere aggiunte per supportare requisiti abbastanza complessi da rappresentare con gli elementi messi già a disposizione. Le quattro categorie di elementi sono:

- Flow Objects
- Connecting Objects
- Swimlanes
- Artifacts

La modellazione dei processi aziendali è usata per comunicare un'ampia varietà di informazioni a differenti ascoltatori. BPMN è progettato per coprire molti tipi di modellazione e permette la creazione di segmenti di processi, così come di interi processi aziendali, a differenti livelli di fedeltà.

2.3. Social BPM

L'avvento delle tecnologie social hanno trasformato il Web da semplice piattaforma dove è possibile reperire informazioni a posto dove gli utenti contribuiscono attivamente alla produzione dei contenuti e di opinioni. La prima generazione di strumenti social, come i blog, wiki e applicazioni per la condivisione dei contenuti, è stata seguita da una seconda generazione di soluzioni, improntati principalmente sulle interazioni interpersonali: i social network sono destinati alla gente comune per incontrarsi virtualmente e stabilire una rete di relazioni sociali. Esistono molte varianti di questo concetto generale, da piattaforme generali come può essere Facebook, ad applicazioni social basate sulla localizzazione geografica come Gowalla e Foursquare, a reti con scopi specifici come Plaxo e LinkedIn.

In questo contesto si sviluppa il Social BPM, che fonde la logica del BPM con il software Social. Il suo scopo è quello di migliorare le prestazioni aziendali attraverso la partecipazione controllata di soggetti esterni al processo di progettazione e di esecuzione delle attività.

L'obiettivo dell'estensione sociale del BPM è quello di migliorare l'efficienza dell'organizzazione e, a seconda del contesto e del tipo di processo, tale obiettivo può essere conseguito in diversi modi:

- Sfruttando i legami tra le persone e le conoscenze implicite dell'impresa, per migliorare l'esecuzione delle attività e la diffusione della conoscenza, in modo da favorire il sostegno reciproco tra gli utenti.
- Aumentando la trasparenza e la partecipazioni alle procedure decisionali, in modo da aumentare la consapevolezza dei processi e l'accettazione dei risultati e, quindi, aumentare la possibilità di raccogliere feedback che possono contribuire al miglioramento dei processi.
- Coinvolgendo tutti i dipendenti al kick-off di un'attività, assegnando l'esecuzione ad un insieme più ampio di attori o trovando l'attore più appropriato all'interno del gruppo di lavoro.

Nel classico BPM, i processi sono definiti a livello centrale da parte dell'organizzazione e distribuiti ai vari esecutori, in questo modo, la partecipazione di questi ultimi al progetto, avviene solo in corso d'opera. Questo approccio può essere migliorato mediante caratteristiche sociali a diversi livelli:

- Participatory Design – è un approccio alla progettazione che cerca i coinvolgere attivamente molteplici attori, compresi gli utenti finali.
- Participatory Enactment – effettua delle fasi di socializzazione, dalla progettazione alla realizzazione, permettendo ad un gruppo limitato di utenti interni di ottenere visibilità e partecipazione al processo. Essi sono diversi dagli utenti che parteciperanno alla fase di implementazione che a loro volta parteciperanno indirettamente al processo tramite l'intermediazione di messaggi e documenti.
- Social Enactment – allarga ancora di più l'esecuzione del processo, consentendo la partecipazione di osservatori esterni.

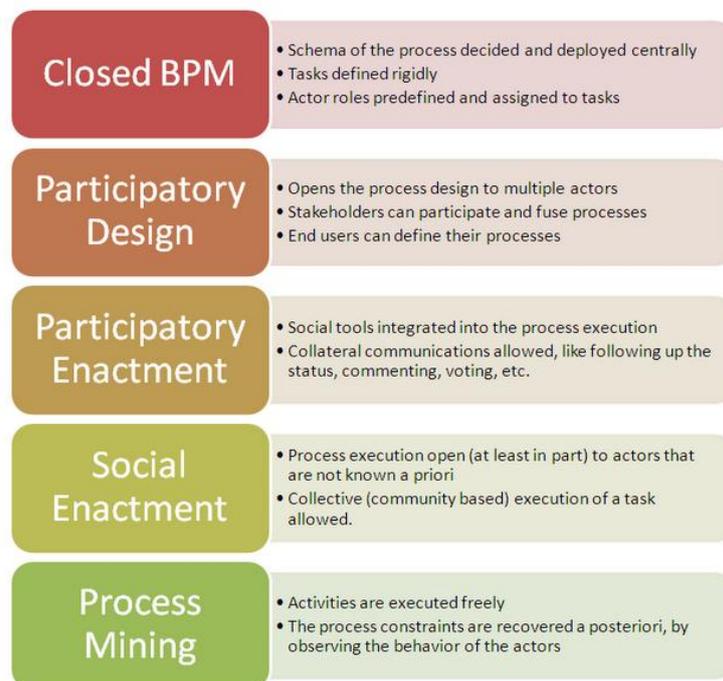


Figura 2.1 – Estensioni del Social BPM

Per coprire questi nuovi aspetti è necessario introdurre nuovi requisiti alla notazione Business Process e agli strumenti di sviluppo, che si adattano bene per descrivere i classici modelli di processi di business, ma sono insufficienti nel descrivere gli aspetti sociali e le interazioni informali tra le persone.

2.4. Soluzioni enterprise per il Social BPM

In questa sezione presentiamo alcune soluzioni sviluppate da importanti aziende del settore dell'IT finalizzate all'integrazione di servizi Social con le tecnologie del BPM per la gestione di processi aziendali; in particolare analizzeremo due soluzioni quali IBM Blueworks Live e Intalio Social BPM che soddisfano pienamente i principi del Social BPM e favoriscono la definizione di processi aziendali integrati in contesti sociali.

2.4.1. IBM Blueworks Live

Blueworks Live fornisce, in tempo reale, la piena visibilità di come il lavoro all'interno dell'organizzazione aziendale sia portato a termine. Come manager, è possibile osservare chi sta lavorando su ogni task e in quale punto del processo si è giunti; inoltre ha la capacità di assegnare alle attività nuovi utenti, in modo da far rispettare le deadline assegnate. Come partecipante al processo, è possibile conoscere in anticipo quali task gli saranno assegnati, controllare le attività completate e scegliere di seguire gli eventi che circondano e interagiscono con i suoi lavori. Blueworks Live fornisce il punto di vista ottimale a chiunque abbia bisogno di concentrarsi sui valori aggiunti di ogni attività.

È una comunità sicura e protetta accessibile da con un qualunque browser Internet, senza aver bisogno di installare o mantenere alcun software. Con Blueworks Live, qualunque membro aziendale può lavorare in tempo reale sullo stesso processo da qualsiasi località del mondo; sarà in grado di documentare processi, così come crearli, eseguirli e parteciparvi velocemente e facilmente utilizzando semplici template per checklist o approvazioni.

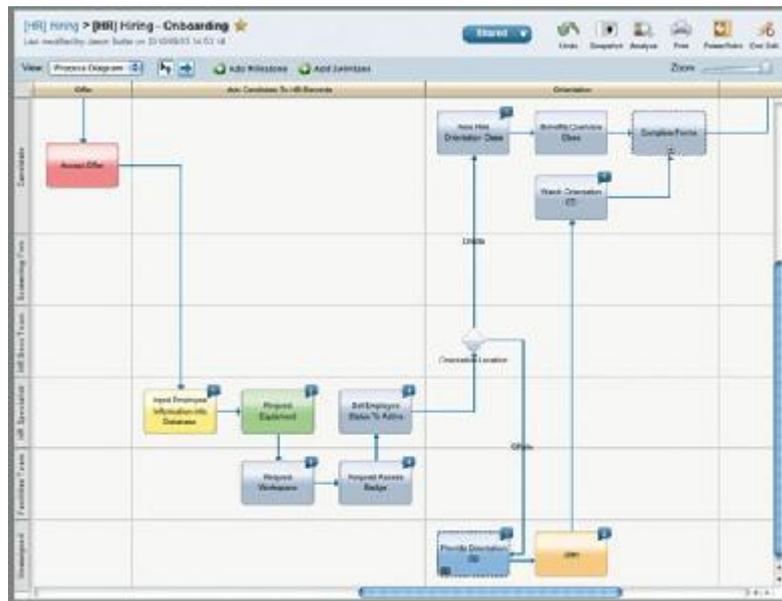


Figura 2.2 – IBM Blueworks Live

Blueworks Live porta con sé un semplice metodo per gestire e documentare processi e innovazioni. È stato progettato tenendo a mente tutti i membri di un'azienda; è abbastanza facile da utilizzare per i neofiti che vogliono costruire una semplice applicazione del processo, ma allo stesso tempo ricco di feature per gli esperti che devono analizzare e modellare anche il più complesso dei processi.

In Blueworks Live si incontrano le opportunità offerte dai social network con le logiche del BPM. Le funzionalità integrate di comunicazione (quali la messaggistica istantanea e la possibilità di commentare il lavoro degli altri) permettono di essere sempre a conoscenza di eventuali modifiche che potrebbero impattare con i progetti che si stanno seguendo. Inoltre costruendo il proprio archivio personale sulle conoscenze organizzative attraverso i processi che si sta modellando e le informazioni che si stanno condividendo, si ottiene accesso immediato alla comunità di esperti di Blueworks Live che possiedono un know-how industriale molto ampio e si occupano di processi, facilitando lo sviluppo di grandi idee per il proprio programma BPM. Con queste avanzate feature di collaborazione, Blueworks Live fornisce all'intero team un modo veloce e più intelligente per migliorare i processi aziendali.

2. BACKGROUND

Gli utenti aziendali avranno un'unica vista attraverso la quale potranno seguire i lavori che gli sono stati assegnati, aggiornare lo stato delle attività delle quali si stanno occupando in quel momento e di iniziare nuovi lavori. Possono completare lo step di un processo completando i campi di una semplice form oppure possono rifiutare la richiesta di lavorare su quel task e possono quindi commentare e riassegnare il lavoro affinché venga garantito il completamento entro i termini previsti. Possono osservare inoltre l'attività dell'intero team che si occupa di quel processo, fare commenti e chiedere aiuto senza che sia necessaria l'assistenza di uno specialista informatico.

I project manager possono configurare ed eseguire semplici processi per il loro gruppo senza l'intervento degli informatici; possono controllare lo stato dei lavori istantaneamente, senza inviare e-mail o effettuare chiamate telefoniche per conoscere il punto della situazione. Ora hanno la sicurezza che il proprio team non mancherà le scadenze grazie al supporto offerto dal sistema di notifiche.

Gli analisti aziendali possono programmare tutti i processi su un singolo sistema e poi condividerli facilmente con il proprio team. Possono usare uno stream interattivo per vedere un riassunto delle modifiche apportate ai processi, visualizzare i feedback degli altri utenti e commentare le operazioni. Possono, inoltre, navigare ed utilizzare template di processi basati sulle migliori tecniche del settore e possono essere sicuri che ogni individuo che dipende da un particolare processo è immediatamente informato di qualsiasi modifica possa essere apportata allo stesso.

I leader di processo COE (Center Of Excellence) hanno una conoscenza dei processi che in passato si poteva avere solo con le capacità manuali ed intellettuali della persona stessa e, con questo tipo di supporto, possono meglio identificare le aree problematiche nei processi. Possono garantire che l'intera conoscenza derivata dai processi all'interno dell'organizzazione sia memorizzata e gestita in un singolo archivio secondo appositi standard e sia sempre aggiornata. Possono facilmente comunicare le migliori tecniche e strategie di processo internamente, mentre possono impararne di nuove dalla larga comunità pubblica e da template di processo predefiniti ampiamente riconosciuti come standard dagli enti del settore.

2.4.2. Intalio Social BPM

Intalio SocialBPM combina le funzionalità offerte da Intalio BPM e Intalio Social per fornire un ambiente centralizzato dove i dipendenti, i clienti e i partner commerciali possono collaborare nella gestione dei processi aziendali attraverso l'intero ciclo di vita.

Le caratteristiche offerte da Intalio SocialBPM sono organizzate lungo sei step del ciclo di vita per la collaborazione sociale, dalla creazione dei contenuti al controllo del sistema. Questa catena garantisce una sicura collaborazione tra i dipendenti, i clienti e i partner commerciali utilizzando strumenti e modelli di comunicazione per i quali possiedono già una certa familiarità nel loro utilizzo.

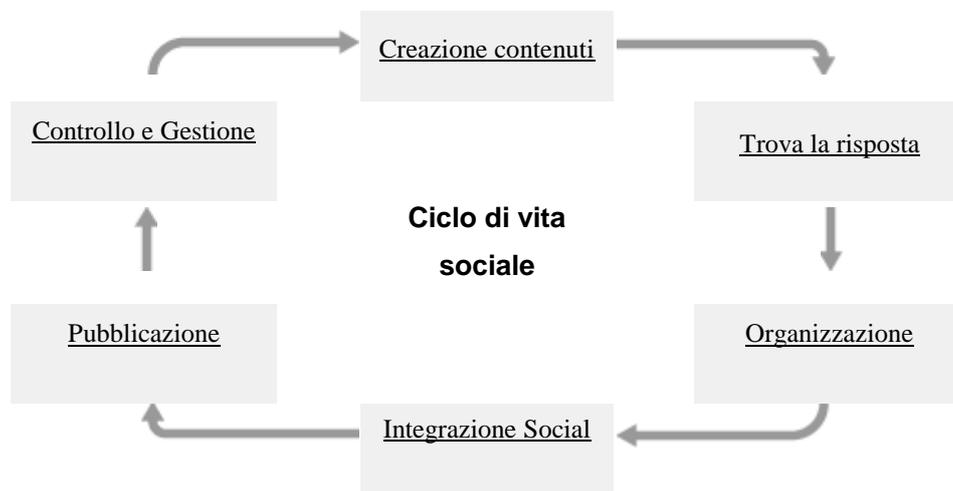


Figura 2.3 – Ciclo di vita Intalio Social BPM

Creazione dei contenuti:

- **Annunci:** invio di messaggi al proprio team, ai clienti e ai partner. Esempi: annunciare le modifiche apportate al processo in produzione.
- **Discussioni:** aprire un topic, o commentarne uno. Fare domande e segnalare quelle che hanno ricevuto una risposta pertinente. Esempi: scoprire processi già esistenti, parlare di idee per il miglioramento dei processi.
- **Blog:** condividere il punto di vista di un utente o di un gruppo di utenti. Esempi: permettere al team di mantenere un log dove registrare le loro attività.

- Wiki: definire una personale enciclopedia per il team. Esempi: documentare processi, procedure e regole in maniera incrementale.
- Sondaggi: procurarsi pareri su uno specifico argomento. Esempi: identificare le possibili aree di miglioramento dei processi, proporre idee ai membri del gruppo.
- Calendario: gestire gli eventi e tracciare la disponibilità degli utenti. Esempi: organizzare i progetti BPM e assegnare task agli utenti in base alla loro disponibilità.
- Documenti: creare documenti in collaborazione o condividere un qualsiasi file memorizzato sul proprio personal computer. Esempi: estrapolare i requisiti per migliorare un processo, allegare documentazione al processo.
- Anteprima documenti: anteprima dei documenti online e possibilità di apporre commenti senza la necessità di effettuare il download sul proprio PC. Esempi: visualizzare e annotare documenti allegati all'istanza del processo in esecuzione.

Trova la risposta:

- Ricerca: ricerche dinamiche attraverso luoghi, utenti e contenuti. Esempi: trovare e riutilizzare componenti di processi già esistenti dall'archivio aziendale.
- Dashboard: configurare la propria home page per tenere sempre sotto controllo tutti gli aggiornamenti interessanti. Esempi: fornire agli utenti un visione d'insieme di tutti i processi aziendali relativi alla produzione.
- Preferiti: osservare i trend più popolari e tenere traccia dei propri preferiti. Esempi: creare collegamenti tra la dashboard di un processo e gli aspetti più interessanti dello stesso e le relative pagine Web.
- Categorie: fornire agli utenti opportune categorie per organizzare al meglio i loro contenuti. Esempi: catalogare i processi secondo unità aziendale, geografia o altra categoria.
- Collegamenti: link verso un qualsiasi oggetto o processo. Esempi: collegare il modello di un processo con la relativa documentazione.

- Tag: etichette personali degli utenti utilizzate per catalogare i contenuti. Esempi: sviluppare un'ontologia per regole e processi aziendali.

Organizzazione:

- Spazi: creazioni di spazi di collaborazione per qualunque oggetto come clienti, prodotti o progetti. Esempi: favorire la collaborazione tra analisti di processo e specialisti IT.
- Gruppi: organizzare comunità, ovunque e in ogni momento. Esempi: organizzare team BPM distribuiti attraverso unità aziendali e geografiche.
- Progetti: condividere la creazione di task e timeline. Esempi: tenere traccia dei progetti BPM, dalla richiesta al ritiro.
- Archivio privato: si possono avere discussioni e documentazioni private, distinte da quelle di gruppi o spazi. Esempi: garantire la riservatezza di progetti sensibili.

Integrazione social:

- Profili: farsi conoscere dai colleghi e contattare esperti. Esempi: individuare esperti del processo basandosi sulle loro conoscenze.
- Conessioni: rimanere sempre aggiornati sulle attività dei membri del team. Esempi: seguire il lavoro dei colleghi che lavorano su importanti progetti BPM.
- Stato: condividi ciò che stai facendo al momento. Esempi: tenere traccia delle attività di tutte le persone interessate al processo.
- Messaggistica istantanea: chat con i colleghi online. Esempi: discutere dei modelli dei processi in tempo reale, fornire aiuto online agli utenti.
- Aiuto: fornire supporto ai nuovi utenti. Esempi: accelerare e facilitare l'inserimento di esperti e utenti del processo.

Pubblicazione:

- Siti pubblici: rendi la comunità visibile a nuove realtà. Esempi: pubblicare i modelli di processo sulla rete aziendale.

- Portlet Designer: progettare rapidamente nuovi componenti per il portale senza dover scrivere righe di codice. Esempi: estendere le dashboard dei processi e le interfacce dei flussi lavorativi con funzionalità personalizzate.
- E-mail: seguire, rispondere o creare contenuti dal proprio dispositivo mobile. Esempi: mantenere aggiornata le attività del processo al volo.
- SEO: aumentare traffico verso il sito utilizzato parole-chiave generate dai contenuti degli utenti. Esempi: migliorare la precisione e l'accuratezza delle ricerche utilizzando l'archivio aziendale.

Controllo e Gestione:

- Controllo del profilo: definire quali parti del profilo rendere visibili per proteggere la privacy dei membri. Esempi: dare agli utenti pieno controllo del loro profilo pubblico.
- Ambiente sicuro: amministrazione basata sulla distinzione dei ruoli e un controllo degli accessi minuzioso permettono di dare privilegi solo a chi si desidera veramente. Esempi: assicurare che il corretto accesso venga concesso solo alle giuste persone interessate.
- Architettura scalabile: supportare le esigenze di più organizzazioni sulla stessa infrastruttura. Esempi: fornire a più divisioni e unità aziendali accesso alla piattaforma BPM.
- Opzioni di deploy multiple: offrire sia una licenza con servizio di host integrato sia senza. Esempi: mantenere i processi critici in casa, usare la piattaforma cloud in modo appropriato.
- Architettura a plugin: personalizzare i propri rilasci con l'utilizzo di una facile architettura a plugin. Esempi: estendere la piattaforma BPM con applicazioni e componenti di terze parti.
- Interfaccia di gestione del sistema: integrazione con applicazioni e strumenti di gestione preesistenti. Esempi: gestire la piattaforma BPM con gli strumenti esistenti, continuando a seguire le migliori tecniche esistenti.
- Supporto multi-lingua: ricerca multi-lingua e analizzatori adatti ad ogni idioma. Esempi: distribuire i processi aziendali in tutta l'organizzazione.

2.5. Personal Process Management

Nel mondo accademico la discussione sul Personal Process Management è stata poco trattata perciò di lavori che parlano di questo argomento ne esistono ben pochi.

Tra i più interessanti c'è un Technical Report dell'Università del Nuovo Galles del Sud [8] nel quale viene discussa una possibile implementazione per la gestione dei processi personali; in questo lavoro è stato proposto di creare delle strutture per le proprie attività come faremo anche noi successivamente ma ci sono alcuni aspetti fondamentali che differenziano la loro idea dalla nostra. In primo luogo la scelta dei costrutti da utilizzare: hanno trattato solo costrutti sequenziali e condizionali, evitando quindi quello parallelo; in secondo luogo la loro soluzione è destinata principalmente all'uso personale senza la possibilità di condividere ed assegnare task ad altri utenti. La scelta di quali costrutti utilizzare è in qualche modo correlata a quest'ultimo aspetto infatti l'utilizzo del parallelo impone che, di fatto, due attività vengano svolte nello stesso periodo temporale ma, avendo indirizzato il proprio studio sul singolo utente, questa situazione è difficile che si verifichi nella realtà dato che un singolo individuo può concentrarsi solo su un'attività per volta. Inoltre hanno proposto una grammatica formale per la definizione dei processi e ciò rende questo lavoro meno appetibile per l'utente medio e lo lascia ristretto ad esperti del settore mentre noi, cercando di indirizzare la nostra applicazione verso un pubblico di non esperti dell'ambiente, abbiamo scelto di sfruttare i vantaggi della modellazione grafica per la creazione delle nostre strutture: questo fatto, se da un lato offre meno robustezza e formalità dall'altro si propone di essere più semplice ed intuitivo.

Sulla scelta di utilizzare la modellazione grafica, si è espresso in un articolo anche l'autore di questo blog [9] illustrando i vantaggi che questa tecnologia porta con sé: gli elementi grafici non dovrebbero essere nuovi all'utente perché molto probabilmente li ha già utilizzati in qualche altro contesto, anche scolastico, come i diagrammi di flusso ed inoltre si sostituiscono ad una serie di termini tecnici e regole formali appartenenti ad un qualche linguaggio testuale che chiunque avrebbe dovuto conoscere se avesse voluto utilizzare tale soluzione; si risparmia tempo in questo modo perché la fase di

apprendimento è notevolmente accorciata e non si obbliga l'utente ad un ulteriore lavoro di studio e memorizzazione.

Anche Michael Rosemann nel suo blog personale [7] si augura e al tempo stesso propone di sfruttare i principi del BPM per le attività quotidiane di ognuno di noi; questo perché ha notato come alcuni eventi nella vita di una persona possano facilmente paragonarsi a dei processi aziendali come, ad esempio, l'acquisto di una casa o l'organizzazione di un matrimonio e come al tempo stesso i processi aziendali discendano direttamente dalle normali attività quotidiane della gente come ad esempio l'accensione di un mutuo per l'acquisto di una casa.

Questi sono i principali aspetti trattati privatamente da esperti del settore e dal mondo accademico in generale: non possiamo non notare, quindi, come i processi personali siano stati molto poco studiati e approfonditi, ed è proprio in questo contesto che noi cerchiamo di colmare quella distanza tra il mondo aziendale e la vita privata delle persone proponendo una soluzione che potrebbe aiutarle a rendere più efficiente la gestione delle proprie attività come è già accaduto per le imprese con l'automatizzazione dei processi.

3. Approccio

3.1. Introduzione

Abbiamo visto come le soluzioni esistenti per la modellizzazione dei processi siano destinate principalmente verso ambiti aziendali. Per questo motivo il semplice utente si trova disorientato nell'utilizzare tali strumenti per organizzare le proprie attività o raggiungere i propri obiettivi eseguendo una serie di task logicamente legati tra loro in successione: la presenza di molti strumenti di modellazione rende la definizione di una semplice soluzione assai più difficoltosa dato che dovrà per forza seguire un approccio aziendale nella risoluzione del suo problema quando, invece, vorrebbe avere degli strumenti immediati e di facile apprendimento e utilizzo.

Questa situazione costringe l'utente a ricondursi alle soluzioni presentate nella prima sezione del capitolo precedente, cioè a quel tipo di strumenti attraverso i quali è possibile annotare i propri task ma non è possibile creare dei legami logici tra gli stessi: viene meno il concetto di struttura o flusso di task e quindi non sarà possibile definire dei percorsi da seguire per raggiungere determinati obiettivi. In queste applicazioni il singolo task rimane fine a se stesso, può in alcune situazioni essere inserito in un progetto ma appartiene comunque ad una lista disordinata e la sua esecuzione o il suo completamento non influenzano minimamente lo stato degli altri task appartenenti allo stesso progetto.

In un post del suo blog personale [7] Michael Rosemann si domanda come sia possibile che gli strumenti del BPM ancora non abbiano incontrato le esigenze dell'utente medio nella gestione del suo processo naturale di vita dove possiamo ricondurre alcuni eventi a dei veri e propri processi come l'acquisto di una casa o l'organizzazione del suo matrimonio. Si chiede anche il perché le aziende non ci abbiano pensato in quanto la maggior parte dei processi da loro trattati siano dovuti essenzialmente ai processi in vita dell'utente: ad esempio è immediato osservare che non avremmo il processo per l'accensione di un mutuo per l'acquisto di una nuova

casa se non esistesse il processo di acquisto di una nuova casa da parte dell'utente stesso.

Un lavoro dell'Università del Nuovo Galles del Sud [8] illustra la definizione di un linguaggio attraverso il quale definire i processi personali e successivamente modellarli con un'opportuna rappresentazione grafica del linguaggio stesso. La soluzione proposta si avvicina molto al lavoro svolto in questa tesi tuttavia ci sono alcune caratteristiche rilevanti che lo rendono completamente differente: innanzitutto in [8] manca completamente la parte di integrazione con i social network perché si occupano di definire processi personali i cui task possono essere eseguiti solamente dall'utente e, quindi, manca una procedura di condivisione o di assegnamento di attività ai propri contatti; mancando sotto questo aspetto è giustamente assente anche il concetto di parallelismo, cioè di attività che possono essere svolte in parallelo, in quanto si assume che l'utente non sia in grado di eseguire più di un'operazione per volta. Altra differenza con il lavoro che proponiamo è il metodo di definizione del processo: in [8], questa operazione è svolta tramite l'inserimento di informazioni testuali il che non è adatto se il nostro target di riferimento è un utente che ricerca facilità e immediatezza d'uso e non desidera ricorrere a complicate espressioni testuali e a sottostare a precise regole grammaticali.

Su quest'ultimo punto, in un articolo online [9], l'autore spiega i benefici che potrebbe avere se sfruttassimo i concetti della programmazione grafica nella fase di modellizzazione dei processi. Operando con elementi grafici l'utente non si troverebbe più disorientato di fronte all'uso di termini tecnici e di precise regole formali per definire il suo personale processo avendo già conosciuto i classici simboli utilizzati in altri contesti come, per esempio, nella rappresentazione di un diagramma di flusso. Oltretutto l'utilizzo di una rappresentazione grafica per codificare i processi è favorita dalla massiccia diffusione dei nuovi dispositivi hardware quali smartphone e tablet nei quali sta diventando l'interfaccia delle applicazioni il vero aspetto verso il quale concentrare lo sviluppo: un'applicazione ricca di funzionalità ma poco usabile a causa della complessità dell'interfaccia quasi certamente raccoglierà meno consensi di un'applicazione più semplice ma con un'interfaccia che la rende di facile apprendimento e di immediato utilizzo.

Il nostro lavoro cerca quindi di mettere insieme tutti gli aspetti sopra riportati per aiutare l'utente finale nella creazione dei suoi processi personali in modo facile ed intuitivo: sfruttando i principi della programmazione grafica daremo la possibilità all'utente di definire gli schemi che definiscono il suo processo e di assegnare i task che lo compongono ai contatti; garantiamo così l'interoperatività tra social network e concetti legati al mondo BPM per utenti semplici dato che finora questi aspetti non erano mai stati presi in considerazione per applicazioni al di fuori del contesto aziendale. Forniremo quindi all'utente finale una soluzione per definire processi relativi alle sue attività quotidiane od occasionali.

3.2. Modello concettuale

La mappa concettuale dell'approccio adottato, rappresentata utilizzando la notazione grafica per i diagrammi ER, è osservabile nella figura sottostante.

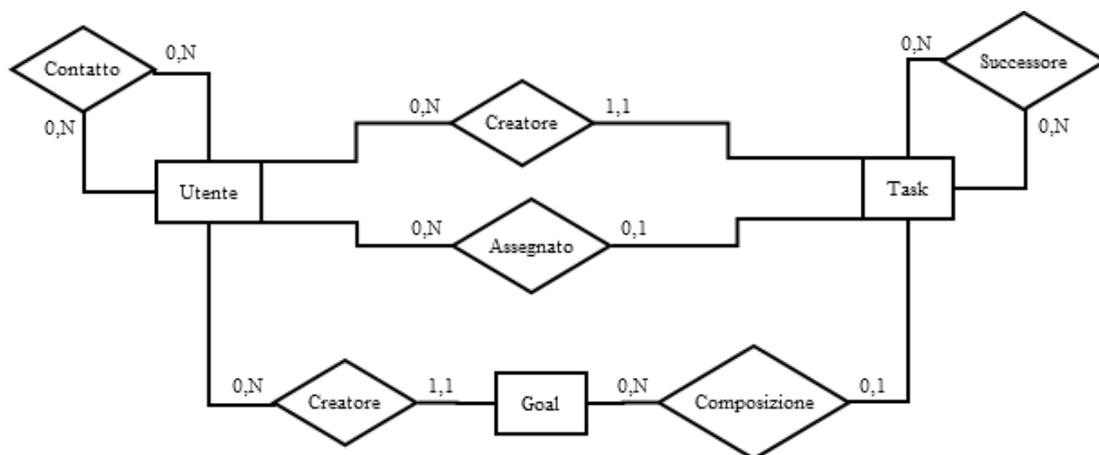


Figura 3.1 – Modello concettuale della soluzione proposta

Come si può ben vedere, i concetti sui quali si basa il nostro lavoro sono estremamente pochi e intuitivamente comprensibili; ciononostante illustreremo in dettaglio ogni elemento del nostro modello nelle prossime sezioni e specificheremo,

dove ciò sia possibile, una formalizzazione matematica per definire quelle proprietà che non possono essere rappresentate da questo schema.

3.2.1. Modello utente

La soluzione proposta in questo lavoro, come abbiamo già detto, è destinata ad utenti non qualificati come esperti del settore dell'Information Technology e quindi i maggiori interessi saranno rivolti alla semplificazione dell'attività di creazione e gestione di processi personali. Questo conduce al non dover utilizzare meccanismi complessi per la definizione dei task o il loro assegnamento ad altri utenti.

Nel nostro sistema l'utente sarà rappresentato da un'entità identificata univocamente da un codice, i cui unici dati strettamente necessari sono i seguenti:

- mail: deve corrispondere ad un reale indirizzo e-mail;
- password: è la chiave segreta dell'utente utilizzata;
- nome: il nome dell'utente;
- cognome: il cognome dell'utente.

I valori di mail e password sono utilizzati per completare il processo di autenticazione dell'utente nel sistema. La coppia di valori nome e cognome servono per facilitare il riconoscimento agli altri utenti del sistema.

Avendo definito gli utenti introduciamo anche il concetto di contatto che corrisponde alla relazione esistente tra due utenti. È normale pensare a questa connessione come una relazione bidirezionale: non è infatti logicamente corretto poter dire che, presi due utenti qualsiasi che chiamiamo $u1$ e $u2$, se $u1$ ha come contatto $u2$, $u2$ non ha, a sua volta, $u1$ come contatto. Formalizzando questo concetto possiamo dire che:

$$\forall u1, u2 \in Utente \quad Contatto(u1, u2) \rightarrow Contatto(u2, u1),$$

dove con *Utente* rappresentiamo l'insieme degli utenti del sistema e con *Contatto(x,y)* la relazione binaria che rappresenta l'esistenza del legame di contatto tra x e y . Questo vincolo è dovuto al fatto che la relazione rappresenta il fatto che due utenti del sistema si conoscono a vicenda e desiderano collaborare insieme.

Osservando la Figura 3.1, si può notare come la partecipazione degli utenti alla relazione contatto non sia obbligatoria ma non abbia neanche un limite massimo; in altre parole, l'utente può avere un numero variabile di contatti, perfino nessuno, anche se in quest'ultimo caso si perderebbero i vantaggi offerti dal contesto social dell'applicazione.

3.2.2. Modello task

Il task è la più piccola unità operativa che compone un processo, può essere eseguita da un unico utente anche se non è il suo creatore: in questo caso è stata eseguita un'operazione di assegnamento ad un utente diverso. Il legame con un processo non è strettamente indispensabile; in questo caso il task equivale ad una semplice attività il cui completamento non dipende dall'esecuzione di altri task.

Elenchiamo ora gli attributi utili per definire un singolo task:

- titolo: identifica il task attraverso una stringa comprensibile all'utente;
- descrizione: descrive l'attività relativa al task in maniera più approfondita;
- stato: indica lo stato del task e può assumere tre valori: *non_visibile*, *visibile*, *completo*;
- deadline: indica il periodo temporale massimo per completare il task;
- start_time: indica il momento in cui il task passa nello stato visibile;
- complete_time: indica l'istante in cui il task passa nello stato completo;

Esistono nel nostro modello alcuni vincoli da rispettare che non sono rappresentati nella Figura 3.1 che esplicheremo tra un momento; servono per mantenere affidabile e logicamente corretto il modello che abbiamo appena descritto.

Innanzitutto ad ogni task deve essere associato un utente che svolge il ruolo di creatore del task; nella modello concettuale questo fatto è rappresentato dalla relazione *Creatore* tra le entità *Task* e *Utente* e dai suoi vincoli di cardinalità: ogni task ha un utente che lo ha creato mentre un utente può aver creato nessuno o infiniti task. In formule questo requisito possiamo tradurlo nella seguente espressione:

$$\forall t \exists u \text{ Creatore}(t, u) \text{ con } t \in \text{Task}, u \in \text{Utente},$$

dove $Creatore(x,y)$ è una relazione binaria la cui semantica corrisponde a “ x è stato creato da y ”.

La relazione $Assegnato$, invece, indica se il task è stato assegnato ad un utente particolare: si assume il fatto che la mancanza di questa relazione per un particolare task t implica che lo stesso task è assegnato automaticamente all’utente creatore; infatti deve essere sempre presente un utente che possa eseguire il task e non deve essere possibile avere dei task che non possono essere completati.

$$\forall t \forall u \text{ Creatore}(t,a) \wedge \neg \text{Assegnato}(t,u) \rightarrow \text{Assegnato}(t,a), \text{ con } t \in \text{Task}, \\ a, u \in \text{Utente},$$

con a che identifica un elemento particolare dell’insieme $Utente$ (cioè il creatore del task t) e $Assegnato(x,y)$ è la relazione binaria che indica l’assegnamento di x ad y .

La proprietà stato del task, come abbiamo già detto, può assumere uno tra i tre possibili valori:

- **non_visibile**: un task si trova in questo stato solo se appartiene ad un processo e tutti i suoi predecessori ancora non si trovano nello stato completo; l’utente infatti non può eseguire questo task se tutti gli elementi precedenti non sono stati completati, nonostante esso sia stato già creato durante la definizione dell’intero processo;
- **visibile**: un task si trova in questo stato se è pronto per essere eseguito dall’utente al quale è stato assegnato; questo stato è impostato di default per tutti i task singoli, cioè che non appartengono ad alcun processo e quindi risultano indipendenti; nel caso di task appartenenti a qualche processo essi risultano visibili solo nel momento in cui i precedenti task sono stati tutti eseguiti e completati;
- **completo**: questo stato è raggiunto solo quando l’utente al quale è stato assegnato il task ha completato l’attività specificata.

Mentre la proprietà *deadline* è un attributo definito in fase di creazione del task che corrisponde all’intervallo di tempo disponibile per portare a completamento l’attività assegnata all’utente, le proprietà *start_time* e *complete_time* definiscono i due istanti

precisi nel quale un task passa nello stato visibile e in quello completo; sono due proprietà aggiuntive che permetteranno di effettuare analisi statistiche sui tempi di esecuzione e controlli se l'avvenuto completamento è in ritardo oppure no rispetto alla deadline prefissata.

3.2.3. Modello processo

Nella mappa concettuale in Figura 3.1, presentiamo il concetto di processo con il nome di *Goal*, cioè esso corrisponde ad un obiettivo per l'utente che l'ha creato; questa legame è rappresentato nella mappa dalla relazione *Creatore* tra le entità *Utente* ed, appunto, *Goal*. Naturalmente un goal è composto da più task (anche nessuno se non è stata ancora definita la struttura associata) mentre il singolo task può partecipare o no alla definizione di un solo obiettivo: la relazione *Composizione* e le relative cardinalità esprimono i concetti appena esposti. Per quanto concerne la struttura, fondamentalmente essa è legata alla relazione *Successore*, creando un nesso logico tra un task e i suoi successori; come vedremo in seguito la struttura di un processo può essere rappresentata da un grafo e con questa relazione non facciamo altro che mantenere le informazioni relative alla lista d'adiacenza di ogni nodo.

Descriviamo ora il modello astratto che rappresenta un processo. Esso è rappresentato da una ennupla $\langle T, E, D, \phi_{Succ}, \phi_{Ass}, \phi_{Temp}, t_{start}, T_{FIN} \rangle$ dove:

- T – insieme dei task che possono concorrere alla composizione di un processo;
- E – insieme degli utenti (esecutori) che eseguiranno i task;
- D – insieme delle possibili deadline, cioè i possibili intervalli temporali, durante i quali è possibile completare il singolo task senza che ciò causi un ritardo.
- $\phi_{Succ}: T \rightarrow \mathcal{P}(T)$ – definisce i successori di un singolo task; può corrispondere all'insieme vuoto e ciò indica l'assenza di un successore per il task o corrispondere ad un sottoinsieme di T ad indicare la presenza di più di un successore.

- $\phi_{Ass}: T \rightarrow E$ – specifica l’assegnamento di un task all’utente indicato per la sua esecuzione.
- $\phi_{Temp}: T \rightarrow D$ – imposta una deadline per il task in oggetto.
- $t_{start} \in T$ – indica il task iniziale del processo.
- $T_{FIN} \subseteq T$ – insieme dei task finali del processo; un task t è detto finale se $\phi_{Succ}(t) = \emptyset$.

Avendo definito una funzione per i successori di un task e considerando l’insieme totale dei task, possiamo rappresentare il processo come un grafo direzionale nel quale i nodi sono rappresentati dai task e gli archi specificano la funzione di successione. Il task iniziale t_{start} è indicato nel grafo con una freccia: nella Figura 3.2 il *Task 1* è il nostro task iniziale. *Task 3*, invece, appartiene all’insieme dei task finali T_{FIN} perché non ha successori.



Figura 3.2 – Esempio di task sequenziali

Il modello che abbiamo sviluppato per i processi personali degli utenti doveva avere ed ha come punto di forza la semplicità: l’utente medio non ha bisogno di particolari strutture per la definizione dei suoi processi e il suo lavoro di modellazione non deve essere gravato da complessi costrutti inutili per il suo scopo finale. Per queste ragioni abbiamo omesso strutture complesse quali la mutua esclusione, cioè la possibilità di seguire due percorsi in alternativa: supponendo di avere ad un certo punto del nostro processo una biforcazione, l’utente avrebbe potuto proseguire solo per uno dei due percorsi disponibili se fosse stata gestita anche la mutua esclusione. Una seconda semplificazione apportata al modello di processo è stata l’eliminazione dei costrutti condizionali. L’opportunità offerta da questi tipo di costrutti è quella di svolgere determinati task o attività solamente se si è verificata una certa condizione ma crediamo che tale supporto sia superfluo in ambienti non aziendali nei quali il tipo di

processi si distanzia molto dai processi aziendali per quanto riguarda il livello di complessità; anche questa scelta è stata quindi operata tenendo sempre a mente come obiettivo finale la facilità d'uso e l'immediatezza per il semplice utente che vuole modellare i suoi processi. L'ultima semplificazione adottata nel modello è l'eliminazione della ripetizione di un task o di una serie di task un numero predeterminato di volte prima di proseguire avanti con le attività inserite nel processo: anche questa soppressione deve essere sempre vista nell'ottica di semplificare le operazioni dell'utente finale durante le operazioni di modellazione del processo.

Come conseguenza del processo di semplificazione precedentemente descritto, a disposizione dell'utente rimangono solamente due classi di costrutti le quali però offrono il giusto compromesso tra espressività e facilità d'uso per i processi personali. La prima classe a disposizione, ed è anche la più semplice, è la classica struttura sequenziale: i task del processo si susseguono linearmente uno dopo l'altro e l'ordine di esecuzione deve rispettare tale sequenza. Infatti un task rimane nello stato di non visibilità fin tanto che il task che lo precede non sia stato completato. Nella Figura 3.2 possiamo osservare un esempio di tale struttura in un processo composto da tre task dove *Task1* rappresenta il task iniziale; al suo completamento verrà attivato *Task2* e via di seguito fino a *Task3*, il cui completamento coincide con la conclusione del processo.

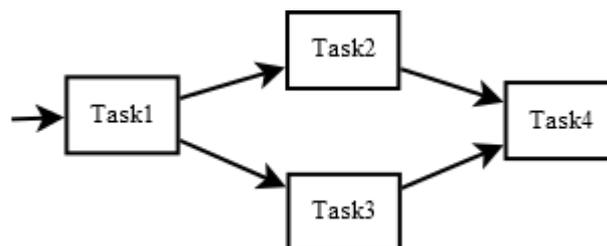


Figura 3.3 – Esempio di task paralleli

Il secondo ed ultimo costrutto messo a disposizione dell'utente è quello relativo al parallelismo; attraverso questa struttura si possono eseguire più task contemporaneamente migliorando, in questo modo, i tempi totali di esecuzione dei processi poiché attività indipendenti l'una dall'altra possono essere eseguite allo

stesso tempo da più utenti distinti. In costrutti di questo genere il completamento di un task può abilitare l'esecuzione di tutti i suoi successori cioè può portarli nello stato di visibilità a meno che questi ultimi non dipendano anche da altre attività non ancora svolte: in pratica un task per essere effettivamente abilitato deve avere tutti i predecessori completati e non solo alcuni di essi.

$$\forall t, t_1 \text{ Predecessore}(t_1, t) \wedge \text{Completo}(t_1) \leftrightarrow \text{Visibile}(t), t, t_1 \in \text{Task},$$

dove $\text{Predecessore}(x, y)$ indica che x è un predecessore di y .

Nella Figura 3.3 si può osservare un esempio di tale costrutto nel quale Task2 e Task3 possono essere eseguiti parallelamente ma solo dopo l'esecuzione del task iniziale Task1 ; quando entrambi saranno completi, Task4 si porterà nello stato di visibilità e l'attività corrispondente potrà essere svolta dall'utente al quale era stata assegnata concludendo l'esecuzione del processo.

È ovvio che l'utilizzo di queste due classi di costrutti può essere combinato affinché sia possibile definire e modellare processi personali che richiedono un livello più alto di espressività rispetto alle semplici strutture sequenziali o parallele; si possono avere, quindi, strutture miste con diversi livelli di annidamento per descrivere meglio anche le attività più complesse ma, nonostante questo, la disponibilità di solamente due tipi rende semplice ed immediata la fase di modellizzazione per gli utenti che non sono esperti del settore. Anche graficamente, come poi vedremo, la comprensione del funzionamento e dell'esecuzione del processo non risulta complicata dalla presenza di elementi inusuali nell'ambiente consumer ma sempre disponibili nelle soluzioni enterprise per l'ambito aziendale.

3.2.4. Controllo di ciclicità

La creazione di un processo attraverso gli strumenti grafici e il suo stretto rapporto con la teoria dei grafi ha fatto sì che si mettesse in luce un problema che potrebbe incidere sulla corretta esecuzione di un processo: la presenza di cicli tra i task che lo compongono. Abbiamo già detto, infatti, che per motivi di semplicità nel nostro modello di processo sono stati eliminati i costrutti necessari per la definizione di operazioni cicliche; inoltre per un corretto funzionamento sarebbe stato necessario anche un operatore condizionale che permetta un controllo sulla condizione di uscita

dal ciclo. È quindi necessario notificare all'utente che sta definendo una sequenza ciclica all'interno del processo altrimenti si corre il rischio di avere dei loop i quali impediscono di portare a compimento l'intero processo.

Iniziamo col dire come riconoscere un ciclo data la definizione formale del processo fornita nella sezione precedente; abbiamo bisogno di una funzione che attribuisca ad un task t l'insieme dei task che possiamo raggiungere seguendo tutti i possibile percorsi del grafo: ciò corrisponde alla chiusura transitiva della funzione ϕ_{succ} che indicheremo con $\phi_{succ}^*: T \rightarrow \mathcal{P}(T)$. A questo punto, se lo stesso task t compare nell'insieme dato da $\phi_{succ}^*(t)$ vuol dire che siamo in presenza di un ciclo perché attraverso un possibile percorso da t riusciamo nuovamente a raggiungere t . Più precisamente:

$$\nexists \text{ un ciclo} \Leftrightarrow \forall t \in T \quad t \notin \phi_{succ}^*(t)$$

Con alcuni esempi mostriamo quanto sia semplice e logico il discorso. In Figura 3.4a possiamo osservare il caso più semplice in assoluto, cioè la presenza di un auto-anello.

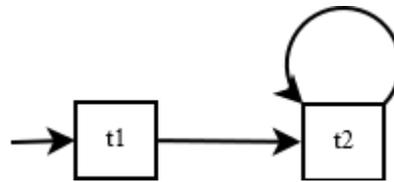


Figura 3.4a – Ciclo: struttura semplice

Se applichiamo la funzione ϕ_{succ}^* ad ogni task otteniamo che:

- $\phi_{succ}^*(t1) = \{t2\}$,
- $\phi_{succ}^*(t2) = \{t2\}$

e quindi un ciclo esiste come si vede anche dalla Figura 3.4a.

Ma anche in esempi più complessi come quello in Figura 3.4b possiamo osservare come la definizione data mantenga la sua validità.

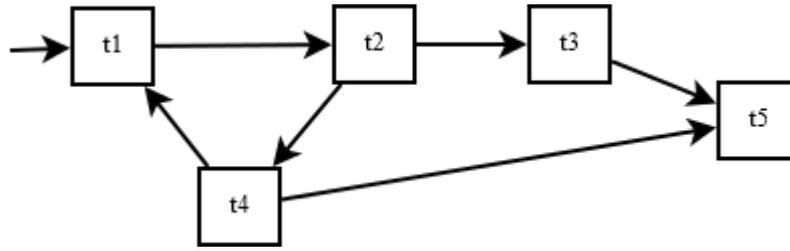


Figura 3.4b – Ciclo: struttura complessa

In questo caso, infatti, avremo:

- $\phi_{Succ}^*(t1) = \{t2, t3, t4, t5, t1\}$,
- $\phi_{Succ}^*(t2) = \{t3, t4, t5, t1, t2\}$,
- $\phi_{Succ}^*(t3) = \{t5\}$
- $\phi_{Succ}^*(t4) = \{t1, t5, t2, t3, t4\}$
- $\phi_{Succ}^*(t5) = \emptyset$

Anche qui quindi esiste un ciclo t1-t2-t4, ognuno di essi infatti ha se stesso tra i task raggiungibili e questo non vogliamo che accada per non influire sulla corretta esecuzione del processo. Nell' ultimo esempio che andiamo ad analizzare mostriamo un processo senza cicli.

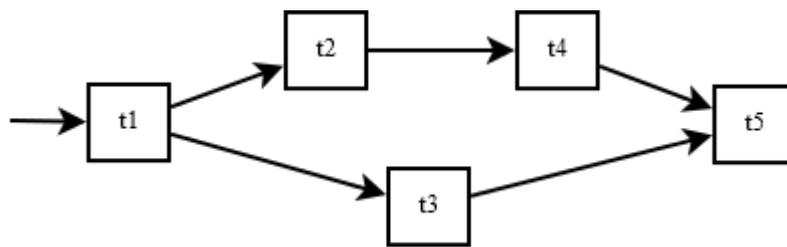


Figura 3.4c – Processo senza cicli

Nella Figura 3.4c, il processo rappresentato non presenta cicli come è anche possibile intuire visivamente; applichiamo però la definizione e osserviamo come ciò sia vero anche in maniera formale.

- $\phi_{Succ}^*(t1) = \{t2, t3, t4, t5\}$,
- $\phi_{Succ}^*(t2) = \{t4, t5\}$,
- $\phi_{Succ}^*(t3) = \{t5\}$
- $\phi_{Succ}^*(t4) = \{t5\}$
- $\phi_{Succ}^*(t5) = \emptyset$

Come abbiamo potuto osservare ogni task che compone il processo non è contenuto nel suo stesso insieme di successione definito da ϕ_{Succ}^* e per questo motivo il processo risulta esente da cicli che porterebbe ad un'errata esecuzione.

Tale tecnica di controllo sarà quindi implementata nella nostra soluzione per impedire all'utente di creare grafi di processo ciclici.

3.3. Integrazione social

Abbiamo visto finora il ruolo dell'utente da un punto di vista prettamente applicativo, cioè non ci interessava sapere realmente chi fosse ma solamente come interagiva col sistema e a quali interazioni prendeva parte: lo abbiamo preso in questo modo senza inserirlo in un contesto sociale. Ma abbiamo già parlato nella nostra introduzione di come gli utenti siano sempre più propensi nell'utilizzare i social network e di come questi li spingano ad instaurare nuove relazioni e a migliorare quelle già esistenti.

Nell'ambito del nostro lavoro, questi rapporti possono rivelarsi molto utili: i contatti di un utente registrato in un social network possono partecipare attivamente all'esecuzione di alcuni task di un processo oppure può l'utente stesso concorrere al completamento dei processi dei suoi amici svolgendo le attività che gli sono state assegnate. Ma il vero vantaggio offerto dal mondo social è la vastità della rete di contatti che si porta dietro: non è più necessario ripetere le solite procedure per l'instaurazione di un rapporto con un altro utente del nostro sistema se questo già esiste in qualche social network. Inoltre si possono sfruttare tutte le funzionalità da loro offerte per lo scambio di messaggi, commenti, opinioni e idee così come si può

utilizzare il sistema di notifiche messo a disposizione per avvertire gli utenti di eventi particolari.

Analizziamo in maniera più dettagliata come la nostra soluzione interagisce con il mondo dei social network.

Innanzitutto l'utente può sfruttare i suoi dati di autenticazione al social network per interagire con il nostro sistema: si evita all'utente l'intero processo di registrazione e quindi l'utilizzo di un'ulteriore nuova coppia di credenziali (mail e password) che dovrà memorizzare per poter accedere. Avendo effettuato l'accesso con le credenziali di un social network, l'utente avrà l'opportunità di poter osservare i contatti provenienti dal profilo social i quali si presenteranno in due possibili categorie:

- il contatto ha già utilizzato il sistema – risulta già inserito e perciò apparirà tra i contatti disponibili per collaborare ad attività e processi: a questo tipo di utenti è quindi possibile assegnare task e compiti da svolgere;
- il contatto non è registrato nel sistema – l'utente può scegliere determinati contatti presenti in questa seconda categoria e inviare loro una richiesta di partecipazione allo svolgimento delle attività inerenti i processi; la richiesta sarà visualizzata sulla bacheca o wall del contatto selezionato, cioè sulla pagina personale nel relativo social network.

Oltre queste caratteristiche, il sistema opererà al fine di interagire attivamente con gli utenti coinvolti nell'esecuzione dei processi in diversi modi:

- ogni qual volta un task del processo passa nello stato visibile, una notifica avvertirà l'utente al quale è stato assegnato il task che c'è un'attività da portare al termine;
- al passaggio di un task nello stato completo, l'utente creatore dello stesso e, quindi, del processo cui appartiene riceverà una notifica dell'avvenuto completamento.

Ovviamente, sfruttando a questo punto le capacità offerte dai social network, è possibile commentare questi eventi e instaurare, ove necessario, anche discussioni per chiarire eventuali dubbi e perplessità.

3.4. Esempi

Esaminiamo ora un paio di esempi di processi personali e illustriamo il modello ottenuto in fase di definizione; premettiamo che nelle immagini sono omesse alcune informazioni per mantenere un adeguato livello di leggibilità e non appesantire troppo le figure con troppi dettagli dei quali parleremo in fase di analisi.

Il primo esempio che portiamo in esame è quello di un utente che desidera organizzare una partita a calcio o calcetto (ma può adattarsi a qualsiasi sport dove si scontrano due squadre).

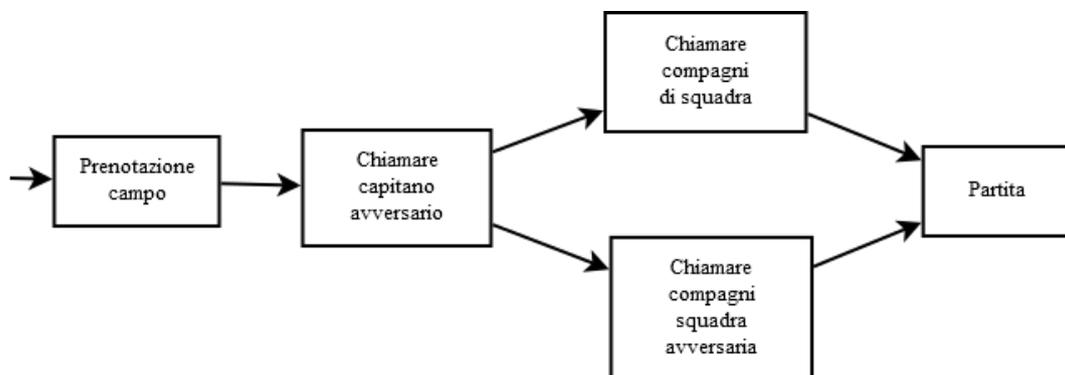


Figura 3.5 – Esempio: organizzazione partita a calcetto

L'utente desidera organizzare una partita: per fare ciò deve trovare un campo disponibile dove giocare e prenotarlo portando a compimento il primo task. Seguendo la struttura del processo, l'utente stesso dovrà avvertire il capitano della squadra avversaria e avvisarlo di costituire il suo team. Proprio in questa situazione possiamo osservare i benefici della nostra soluzione: l'utente, infatti, si occuperà di organizzare il proprio team e, contemporaneamente, il capitano avversario si dovrà comportare

allo stesso modo per chiamare i suoi compagni; in Figura 3.5 è stata omessa questa informazione ma naturalmente i due task in parallelo sono assegnati rispettivamente ai due capitani delle squadre. Solo quando entrambi gli utenti hanno portato a compimento i propri doveri, è possibile giocare la partita e di conseguenza completare l'esecuzione del processo.

Il secondo esempio che andiamo ad illustrare è il tipico scenario dello svolgimento di un progetto assegnato a gruppi di due o più persone in ambito scolastico o universitario; la stessa situazione si potrebbe verificare in contesti aziendali ma come abbiamo già detto non è questo il nostro target di riferimento.

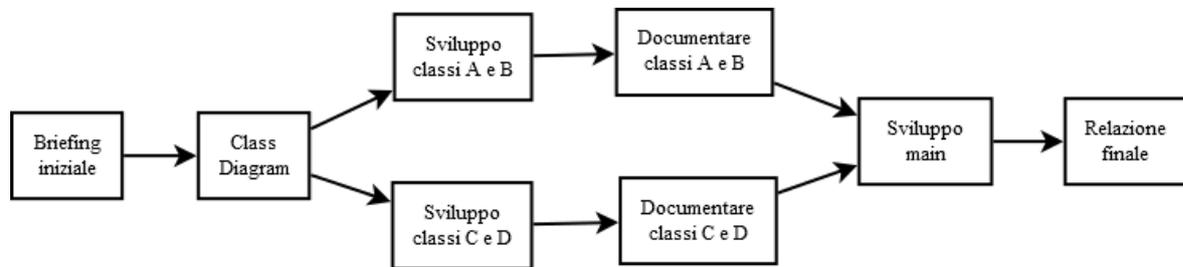


Figura 3.6 – Esempio: progetto scolastico o universitario

Ipotizziamo l'assegnazione di un progetto del corso di programmazione ad oggetti a due studenti, uno dei quali assumerà l'incarico di supervisore e organizzatore delle attività richieste: può per questo compito sfruttare le capacità di modellazione dei processi per distribuire il lavoro tra se e il suo compagno di progetto, automatizzando alcuni compiti; in questo esempio, i due task iniziali saranno svolti in sequenza ma sarà lo stesso supervisore a decidere quando si può ritenere l'attività completata e passare alla successiva. Giunti alla fase di sviluppo, i due studenti possono proseguire il proprio lavoro indipendentemente dallo stato in cui si trova l'altro fino a che non si dovrà sviluppare la parte di codice comune relativa al main: per arrivare a questo task è, però, necessario che entrambi abbiano svolto il lavoro che è stato assegnato loro. Possono infine proseguire con gli ultimi due compiti e concludere positivamente il lavoro sul progetto, come specificato nel modello del processo riportato in Figura 3.6.

Trattandosi di esempi abbiamo omissis nelle illustrazioni anche le informazioni relative alle deadline di ogni task: ovviamente, in fase di definizione del processo, questi vincoli temporali devono essere stabiliti affinché vengano rispettate le eventuali scadenze. Nei progetti universitari, per esempio, è molto comune avere delle date limite oltre le quali non è più possibile consegnare i lavori e per questo motivo il supervisore dovrà scegliere opportune deadline al fine di rispettare i vincoli temporali imposti.

4. Implementazione

In questo capitolo andremo ad illustrare come è stata effettivamente implementata la soluzione proposta nel capitolo precedente seguendo i principi del modello di sviluppo a cascata, uno delle più diffuse strategie di sviluppo adottate e studiate dall'ingegneria del software.

4.1. Analisi dei requisiti

Il primo passo da compiere nello sviluppo dell'applicazione è l'analisi dei requisiti. Ci occupiamo in questa fase di descrivere le principali caratteristiche richieste dalla nostra soluzione immaginando dei possibili scenari d'uso; con questa procedura abbiamo potuto raggruppare le feature in cinque macro aree elencate di seguito:

- autenticazione utente;
- inserimento e completamento task;
- goal (obiettivo);
- gestione contatti;
- integrazione social network.

Nelle prossime sezioni analizzeremo singolarmente ognuna di esse approfondendone alcuni dettagli.

4.1.1. Autenticazione utente

L'utente deve potersi registrare nel sistema fornendo un indirizzo mail valido al quale verrà associato il proprio profilo, una password per impedire un accesso non autorizzato ed un nome e cognome che saranno utilizzati per favorire il riconoscimento tra utenti distinti del sistema nella gestione dei propri contatti. La mail

funzionerà da chiave primaria quindi ad ogni indirizzo mail è possibile associare al massimo un profilo. Effettuata la registrazione, l'utente verrà riportato alla propria home. Se già in possesso dei dati d'accesso (mail e password) naturalmente può accedere alla propria home saltando la fase di registrazione. In questa fase si dovranno gestire principalmente tre tipologie di errore:

- mancato inserimento di tutti i campi durante le registrazione;
- indirizzo mail già utilizzato da qualche altro utente;
- dati di accesso non validi durante la fase di autenticazione.

4.1.2. Inserimento e completamento task

Uno dei due aspetti fondamentali del sistema è l'inserimento dei task che può avvenire in due modi distinti ma collegati tra di essi. Il primo metodo è un inserimento rapido che consiste nell'indicare solamente l'oggetto del task evitando di inserire tutti i dettagli quali, ad esempio, una deadline, una descrizione, a chi assegnarlo; l'utente quindi dovrà semplicemente inserire un titolo e con la semplice pressione di un tasto avrà creato un task con tutti i parametri omessi impostati a dei valori di default. Il secondo metodo di inserimento consente, invece, di specificare tutti o solo alcuni parametri relativi al task: l'idea è quella di associare un link "dettagli" affianco al campo utilizzato per indicare l'oggetto del task che permette la comparsa di una form nella quale si possono individuare i seguenti attributi:

- deadline: termine ultimo per il completamento del task;
- descrizione: un eventuale descrizione informale dell'attività che dovrà essere svolta (default = vuoto);
- assegnato a: permette di identificare un utente al quale assegnare il task tramite il suo indirizzo mail (default = creatore task);

È ovvio comunque che un utente può modificare questi valori in un secondo momento andando a visitare le proprietà del task. In questa fase, naturalmente, l'aspetto più delicato da tenere sotto controllo è l'inserimento di task con il campo oggetto vuoto,

cioè non è possibile creare attività con un titolo vuoto: tutto questo per impedire di avere poi delle liste di task vuoti che all'apparenza sono tutti identici ma magari hanno attributi tutti distinti. Per gli altri campi, come abbiamo già detto, utilizzeremo dei valori di default. Se un utente può inserire dei task, conseguentemente deve avere anche la possibilità di marcarli come completati: l'idea è quella di inserire una checkbox di fianco ad ogni task che se selezionata indica che l'attività è stata completata con successo; per dare maggiore risalto all'azione si può pensare anche di cambiare lo stile del titolo del task, magari sfumandolo su un grigio oppure barrandolo come si farebbe nella realtà depennandolo da una lista su carta.

4.1.3. Goal (obiettivo)

L'innovazione che vogliamo portare in questo tipo di applicazioni è proprio la possibilità di creare delle strutture di task per portare a termine determinati obiettivi con l'interazione di più di un utente ma con alcuni vincoli prestabiliti: vogliamo fornire un metodo per creare sequenze di task che a loro volta possono anche essere eseguita parallelamente; così facendo garantiamo lo svolgimento di quelle operazioni tra loro indipendenti e quindi non necessitano di un'esecuzione sequenziale. Osservando l'immagine in Figura 4.1 capiremo meglio il funzionamento di questo sistema.

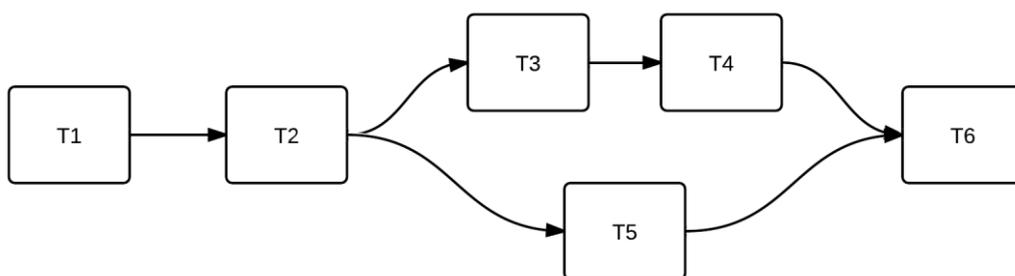


Figura 4.1 – Esempio struttura goal

Come si può vedere, in una struttura di questo tipo, i task T1 e T2 devono essere svolti in successione rispettando l'ordine dato quindi, al completamento di T2, T5 può essere

eseguito contemporaneamente all'esecuzione di T3 e T4; a questo punto, per poter proseguire con l'esecuzione di T6 bisogna attendere il completamento di T4 e T5.

La gestione di questo tipo di strutture può essere effettuata inserendo dapprima i singoli task indipendentemente e poi modellando attraverso opportuni comandi la struttura che conterrà i task stessi: i comandi serviranno a creare liste sequenziali piuttosto che inserire dei nodi necessari per impostare il parallelismo tra task; queste funzioni dovranno essere gestite da un utente che nel caso rivestirà il ruolo di "supervisore dell'obiettivo" in quanto sarà soltanto lui a conoscere l'esatta struttura e quali task compongono il flusso operativo.

I task che andranno a comporre la struttura del goal saranno legati solamente all'obiettivo e quindi non saranno visualizzati come singole attività indipendenti; quindi la fase di definizione della struttura sarà preceduta da un'altra fase durante la quale l'utente inserirà i task dell'obiettivo. Al fine di terminare l'operazione, all'utente non rimane altro che assegnare le attività ai suoi contatti o a se stesso ed impostare le deadline.

4.1.4. Gestione contatti

Dato che vogliamo indirizzare l'applicazione verso un ambito social piuttosto che un ambito business, un efficace gestione dei contatti sarà sicuramente molto importante per il tipo di target-utente scelto. Abbiamo già detto che un task può essere assegnato ad un utente diverso dal suo creatore per questo motivo ora si aprono due possibili scenari: se l'utente destinatario del task è già registrato nel sistema questa operazione non comporta altro che l'aggiunta del task nella sua lista delle attività da completare e, nel caso si tratta di un task relativo ad una sequenza temporale, si può pensare di aggiungere automaticamente un commento nel quale si avverte l'utente che sta concorrendo al raggiungimento di un certo obiettivo; nella situazione in cui il destinatario ancora non è inserito nel sistema, allora è necessario implementare una qualche funzione, ad esempio con una notifica via mail, per invitare l'utente ad iscriversi per completare l'attività che gli è stata affidata ed apportare il proprio aiuto, nel caso il task è parte di una sequenza temporale, al raggiungimento dell'obiettivo perseguito.

L'assegnamento del task avviene, come descritto precedentemente, attraverso l'inserimento dell'indirizzo e-mail dell'utente destinatario tra le proprietà del task stesso. Sebbene questo sistema risulti molto semplice, possiamo pensare ad un sistema alternativo che sfrutti le nuove funzionalità offerte dalla tecnologia informatica: in pratica dopo aver aggiunto un nuovo task con l'inserimento rapido (quindi i suoi campi hanno valori di default ad eccezione dell'oggetto o titolo), trasciniamo il nominativo della persona presente in un apposita lista nell'interfaccia sul task creando un assegnamento "On the Fly". Naturalmente questo sistema, anche se attraente dal punto di vista estetico, presenta lo svantaggio di non poter assegnare task a persone non inserite tra i propri contatti.

La gestione dei contatti deve risultare una funzionalità indipendente: tutte le volte che l'utente utilizzerà il metodo classico per assegnare dei task cioè attraverso il campo tra i dettagli dell'attività, il sistema dovrà tener conto di questi utenti e considerarli come contatti; deve anche offrire una gestione che va oltre al semplice assegnamento e permetta l'aggiunta di nuovi amici tramite mail e la visualizzazione dei dettagli di ognuno (per dettagli in questo caso consideriamo solo nome, cognome e mail). Se l'indirizzo mail è già registrato nel sistema, l'utente associato verrà immediatamente inserito tra i contatti dell'utente operatore in quel momento; nel caso in cui l'indirizzo mail non è associato ad alcun utente, verrà inviata una mail di notifica all'indirizzo inserito con la richiesta di iscrizione al sistema: finché tale richiesta non verrà soddisfatta, la richiesta di "amicizia" sarà lasciata in sospeso.

4.1.5. Integrazione social network

Molti sistemi di questo genere già offrono questo tipo di funzionalità con un elevato tasso di integrazione, nel nostro caso vogliamo limitarci a delle operazioni di base. La prima può essere l'utilizzo dei dati di accesso ad un social network per accedere al nostro sistema evitando, quindi, di dover operare un'ulteriore registrazione, cosa che può risultare alquanto noiosa per la maggior parte dei navigatori del web: si potrà quindi accedere utilizzando i dati del social network già in possesso dell'utente; l'idea principale è quella di integrarlo con un social network tipo Facebook dato che

vorremmo destinare l'applicazione verso normali utenti senza caratteristiche particolari.

Partendo da questo punto, il tasso di integrazione può crescere vertiginosamente se pensiamo, ad esempio, all'importazione dei contatti: nella nostra gestione dei contatti potremmo selezionare tutti o solamente alcuni dei nostri amici presenti sul social network per invitarli all'utilizzo del sistema in modo da averli poi disponibili per l'assegnamento dei task; oppure all'invio di messaggi e/o post in bacheca per notificare al contatto che gli è stata assegnata un'attività da portare a termine. Queste sono solamente alcuni esempi di come l'applicazione possa integrarsi con un social network ma è possibile migliorarsi sfruttando alcune funzionalità avanzate come i tag e i commenti: almeno per il momento, però, ci possiamo limitare alle possibilità appena descritte, anche per mantenere una certa semplicità d'utilizzo.

4.2. Mockup interfaccia

In questa sezione descriviamo la nostra idea di interfaccia utente proponendo alcune bozze per il design finale; visto il nostro scopo di facilitare l'uso di questo strumento all'utente medio, abbiamo posto particolare attenzione verso alcune scelte progettuali finalizzate alla semplicità e all'intuitività. Complessivamente avremo cinque possibili viste per l'utente attraverso le quali può compiere tutte le operazioni desiderate; esse sono:

- registrazione e autenticazione;
- home page utente;
- dettagli task;
- dettagli goal;
- gestione contatti.

4.2.1. Registrazione e autenticazione

La prima schermata (Figura 4.2) mostra la pagina di accesso all'applicazione Web nella quale è possibile effettuare la registrazione o l'accesso al proprio profilo se già si hanno a disposizione i dati per l'autenticazione.

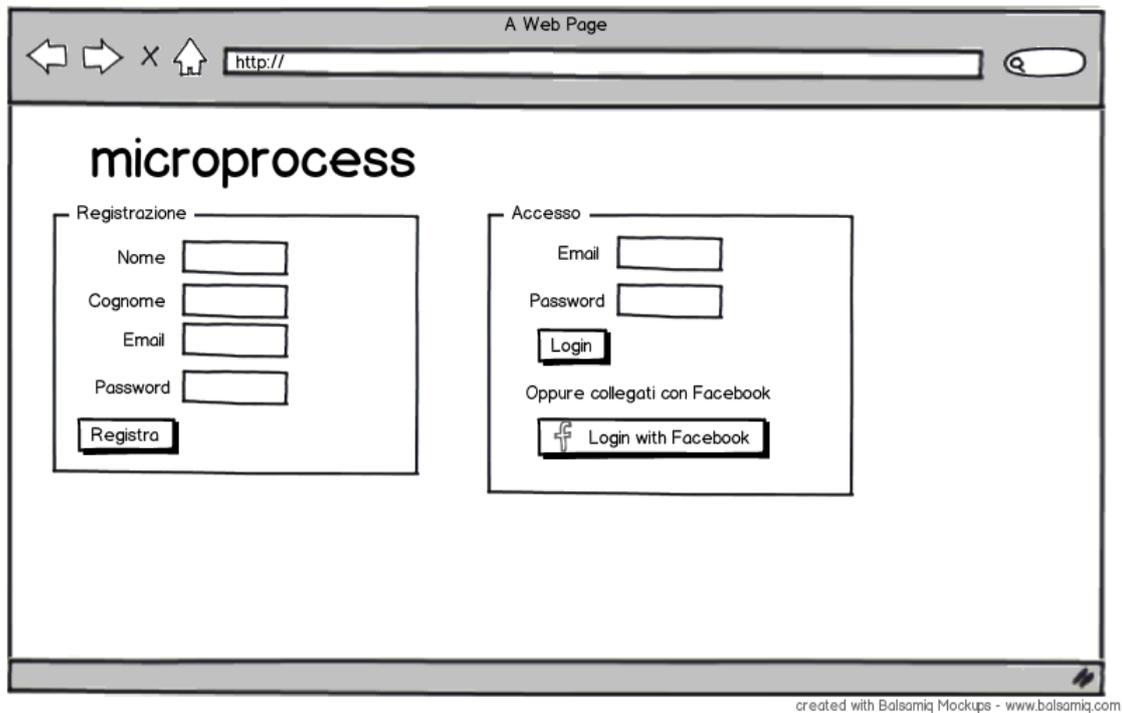


Figura 4.2 – Login e registrazione

La registrazione dell'utente richiede il suo nome e cognome, un indirizzo e-mail valido e una password segreta: questi ultimi due valori sono quindi utilizzati per effettuare l'accesso al sistema nella maniera classica. Ma l'accesso al sistema è consentito anche tramite il processo di autenticazione di Facebook: attraverso questo processo, si evita all'utente la fase di registrazione e l'inserimento dei propri dati per il login. L'unico requisito necessario affinché il sistema operi in maniera corretta è l'autorizzazione all'utilizzo dei dati del social network, richiesta che perviene solamente durante il primo accesso, dopodiché la procedura di login è completamente automatizzata.

4.2.2. Home page utente

Effettuato l'accesso l'utente verrà reindirizzato alla propria home page nella quale sarà visualizzato l'elenco completo dei task suddivisi tra quelli personali e quelli assegnati dagli altri utenti con i quali si è in contatto e gli obiettivi (goal) creati: per ognuno di essi si può accedere alla pagina relativa per la modifica della struttura oppure eliminarlo, inoltre, proprio sotto il titolo, è evidenziato lo stato nel quale il processo si trova al momento dell'accesso dell'utente stesso; i contatti sono elencati nella colonna all'estrema destra della finestra. Sopra l'elenco dei task, è presente il campo per l'inserimento rapido con valori di default: basta semplicemente specificare l'oggetto dell'attività. Cliccando sulla checkbox di fianco al titolo del task lo si indica come completato e contemporaneamente viene aggiornato lo stile con il quale è visualizzato a schermo; alla fine del titolo sono visualizzati i due link "dettagli" ed "elimina": con il primo si va alla pagina di visualizzazione delle proprietà del task, con il secondo lo si elimina.

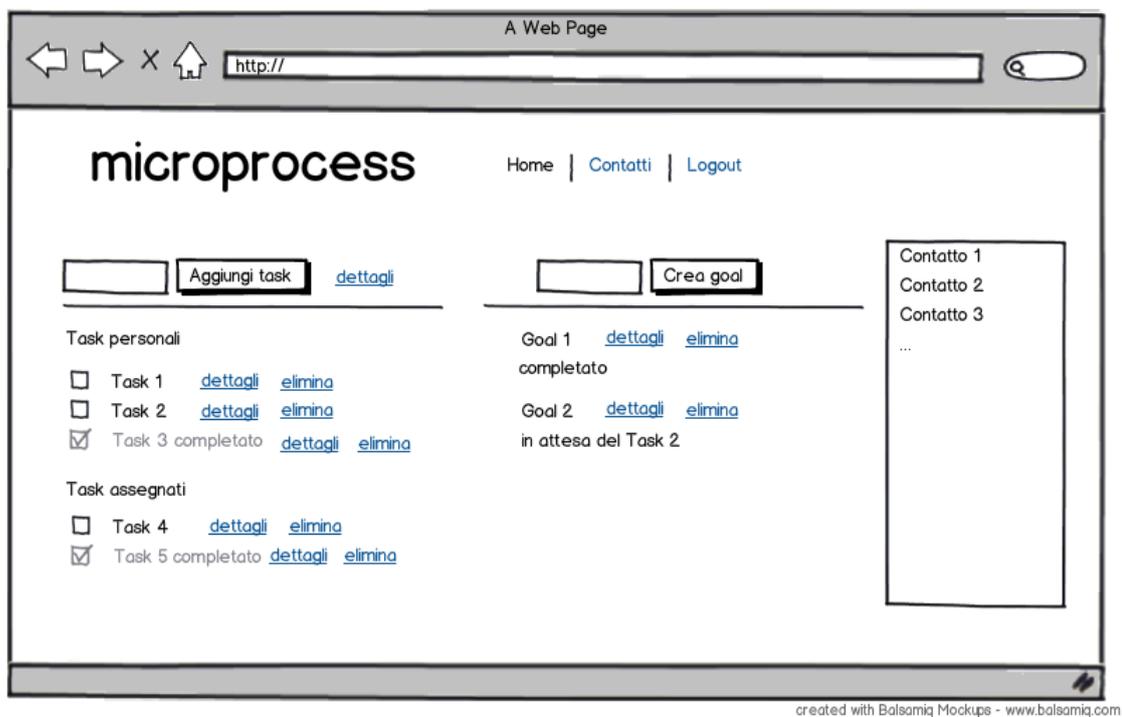


Figura 4.3 – Home page utente

4.2.3. Dettagli task

Eccoci quindi alla pagina dei dettagli del task: a seconda dello stato e della pagina di provenienza, alcuni campi saranno modificabili mentre altri rimarranno già preimpostati; ad esempio, se il task è stato completato ogni proprietà risulterà immutabile. Sarà possibile inserire l'utente assegnatario del task trascinando nel suo il suo nominativo dalla colonna di destra mentre per le deadline si imposterà il valore tramite il menù a comparsa. È anche possibile aggiungere una descrizione opzionale per spiegare meglio l'attività da svolgere.

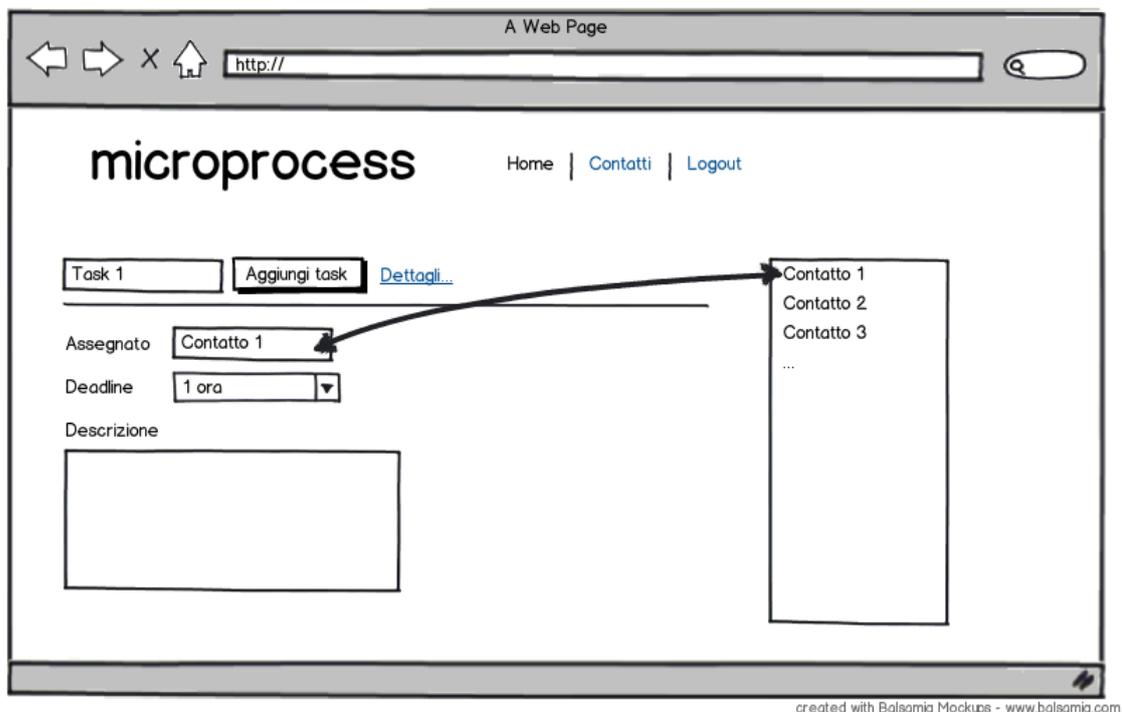


Figura 4.4 – Dettagli di un task

4.2.4. Dettagli goal

Passiamo ora ad esaminare l'interfaccia per quanto riguarda la parte dedicata agli obiettivi cioè alle sequenze strutturate di task tra loro collegati raggiungibile visualizzando i dettagli di un goal nella propria home personale dopo averlo creato.

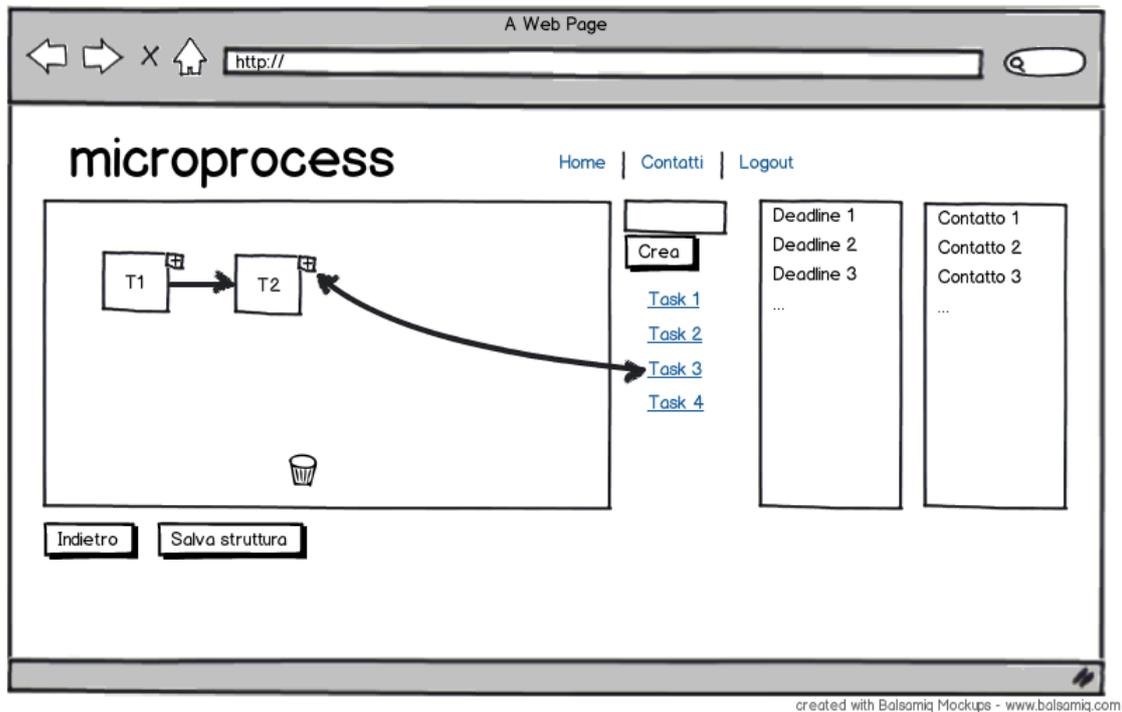


Figura 4.5 – Definizione struttura obiettivo

Vogliamo rendere la definizione della struttura del processo un'operazione relativamente semplice per l'utente: per giungere a questo scopo sfruttiamo la tecnica del drag&drop. Dopo aver creato i task del processo con l'apposita form, l'utente può trascinarli nel pannello dedicato: il primo task corrisponderà all'attività iniziale del processo mentre per aggiungerne di successivi si dovrà trascinarli sul piccolo riquadro contenente il simbolo “+” posizionato accanto al task che si vuole indicare come predecessore del nuovo inserito; ad esempio, in Figura 4.5, se vogliamo aggiungere il Task3 come successore del Task2 è necessario trascinarlo sul “+” associato al riquadro del Task2.

È possibile anche eliminare un task inserito nella struttura: basta trascinarlo verso l'icona del cestino e sarà eliminato il task stesso e tutti i suoi successori che non sono più collegati al resto del grafo, cioè sono diventati irraggiungibili. Terminato l'inserimento, l'utente può definire gli assegnatari e le deadline delle singole attività con la solita tecnica del drag&drop trascinando l'opzione selezionata all'interno del riquadro del task; quando ha terminato la modellazione del processo, l'utente può

tornare alla propria home salvando le modifiche con l'apposito tasto oppure scegliere di annullare tutte le modifiche apportate tramite il tasto "Indietro".

4.2.5. Gestione contatti

In Figura 4.6 mostriamo la pagina di gestione dei contatti dell'applicazione.

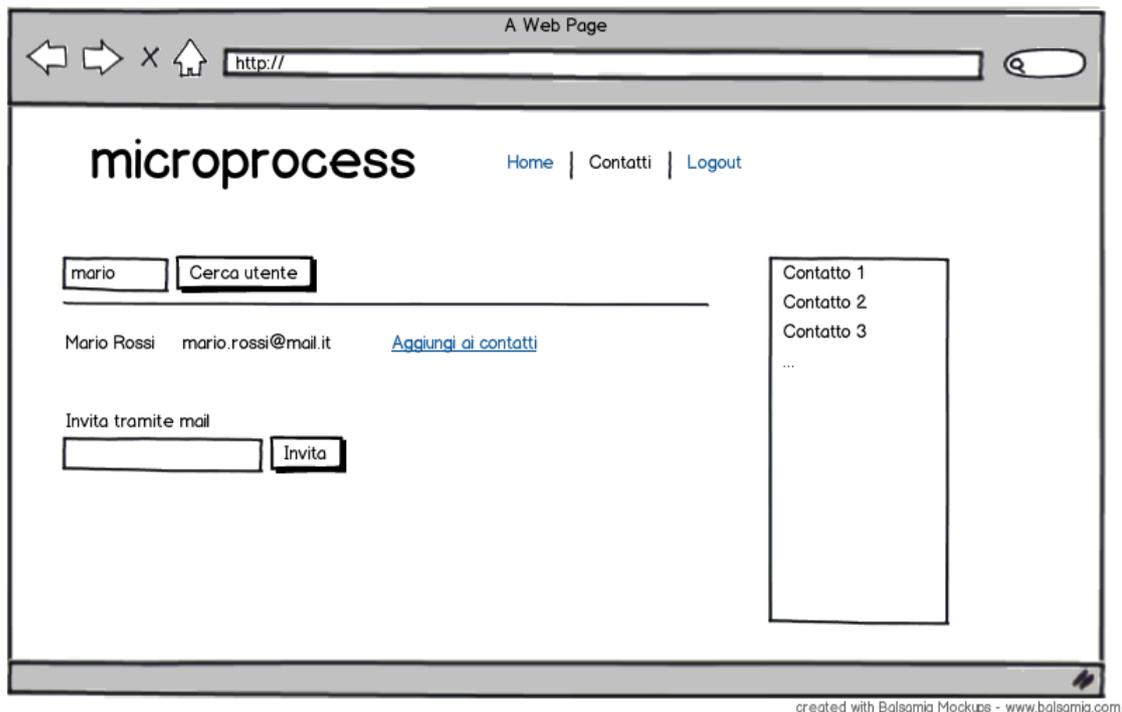


Figura 4.6 – Pagina dei contatti

Come si può vedere è una gestione molto basilare, che racchiude il minimo necessario per permettere poi di avere dei contatti ai quali poter assegnare i task. È possibile ricercare un utente registrato nel sistema immettendo nome, cognome oppure indirizzo e-mail e, nel caso la ricerca abbia esito positivo, vengono visualizzati gli utenti che soddisfano i parametri e accanto ad ognuno di essi è presente un link per aggiungerlo tra i propri contatti. Nel caso la ricerca non produca alcun risultato oppure sappiamo che la persona cercata non è registrata nel sistema, possiamo inviargli una richiesta inserendo il suo indirizzo mail e il sistema provvederà a notificare alla persona l'invito dell'utente con lo scopo di invogliarlo all'utilizzo dell'applicazione.

4.3. Ipotesi semplificative e soluzioni adottate

Durante questa fase di analisi, sono state evidenziate alcune questioni da risolvere e sono state fatte delle ipotesi affinché il funzionamento dell'applicazione risultasse corretto alla fine dello sviluppo della stessa.

4.3.1. Programmazione grafica

Per la definizione delle strutture dei processi sono state proposte più soluzioni ma alla fine la scelta è ricaduta su questa soluzione grafica che crediamo ricalchi le ultime mode in fatto di programmazione web su piattaforme desktop ma che può risultare valido, tramite opportuni accorgimenti, anche su piattaforme mobile dotate di schermi sufficientemente ampi per il suo utilizzo. Inoltre crediamo che questa sorta di programmazione grafica faciliti l'uso di quest'applicazione anche a chi non è esperto del settore come quel target di utenti ai quali noi ci rivolgiamo.

4.3.2. Copia dei processi avviati

Uno dei principali problemi al quale porre rimedio è la modifica della struttura di un processo per il quale è stata già avviata l'esecuzione: può accadere, infatti, che il creatore di un goal voglia modificare la sua struttura dopo che alcuni task che lo compongono sono stati già completati; questi cambiamenti potrebbero non affliggere il corretto svolgimento delle operazioni se, ad esempio, venisse aggiunto un task alla fine della struttura esistente ma nei casi più gravi potrebbero compromettere l'intero flusso operativo del processo. La soluzione adottata per tale problema è la seguente: la modifica è permessa fin tanto che nessun task del goal è stato eseguito mentre è impedita nel caso in cui anche solamente il task iniziale è stato completato: in quest'ultimo caso il sistema dà la possibilità all'utente di annullare le modifiche apportate oppure di creare un secondo processo con una struttura di base identica a quella che si desidera modificare con gli stessi task e le stesse impostazioni per assegnamenti di utenti e deadline.

4.3.3. Assegnamento task ad utenti esterni

La gestione degli assegnamenti è stata semplificata rispetto all'idea originale: inizialmente pensavamo di poter assegnare un task ad un utente inserendo semplicemente l'indirizzo e-mail di quest'ultimo ma ciò costringeva a dover conoscere il giusto indirizzo al fine di impostare l'assegnamento e, poi, era compito del sistema verificare se l'utente fosse già registrato oppure invitarlo ad utilizzare l'applicazione. Ora, invece, abbiamo separato le due fasi: in pratica è possibile assegnare un task solamente ai propri contatti che corrispondono ad utenti già registrati nel sistema. L'invito ad utenti esterni, invece, è rimasta un'operazione a se stante: possiamo invitare una notifica via e-mail oppure se siamo registrati nel sistema con i dati di qualche social network e l'utente è già presente tra i nostri contatti nel profilo social possiamo avvisarlo tramite un post o una notifica sul suo profilo social.

4.4. Design

4.4.1. Architettura

La soluzione proposta si basa su un architettura di tipo client-server tradizionale che sfrutta il canale offerto dal web per la comunicazioni tra i due partecipanti: il ruolo del server è svolto da un server web Apache coadiuvato da un database MySQL che si occupano della gestione dell'applicazione web mentre i client sono rappresentati dai browser che gli utenti utilizzano per interfacciarsi all'applicazione. È necessario inoltre l'utilizzo di un server di posta elettronica per svolgere correttamente le funzioni di invio di inviti agli utenti non registrati nel sistema via e-mail.

Per sviluppare la soluzione abbiamo sfruttato le potenzialità del pattern Model – View – Controller (MVC), cioè abbiamo cercato di implementare separatamente il modello dei dati dall'interfaccia applicativa e gestire il flusso delle informazioni attraverso chiamate a determinate funzioni che ricoprono il ruolo del Controller. Il modello dei dati (Model) è gestito tramite alcune classi che fungono da wrapper per le

informazioni memorizzate sul database MySQL mentre le funzioni del Controller si riconducono nella maggior parte dei casi a chiamate asincrone che partono dai comandi impartiti dall'utente: le risposte possono essere raccolte di dati da visualizzare nell'interfaccia o semplici verifiche sull'effettiva correttezza dell'operazione richiamata; in ogni caso sarà sempre il Controller a reindirizzare l'utente verso la giusta pagina web (View).

4.4.2. Struttura database

Per il progetto del database di supporto abbiamo prima sviluppato un modello concettuale e poi quest'ultimo è stato tradotto in uno modello logico relazionale; nei prossimi due paragrafi descriviamo i due modelli appena citati, poi concludiamo con un'analisi dettagliata degli attributi delle tabelle.

Modello concettuale

Lo schema concettuale del database sottostante l'applicazione riprende le stesse entità e relazioni del modello presentato in Figura 3.1 nel Capitolo 3; ad esse si aggiunge l'entità *Div* per fini puramente implementativi in quanto questo concetto è collegato alla rappresentazione grafica del processo.

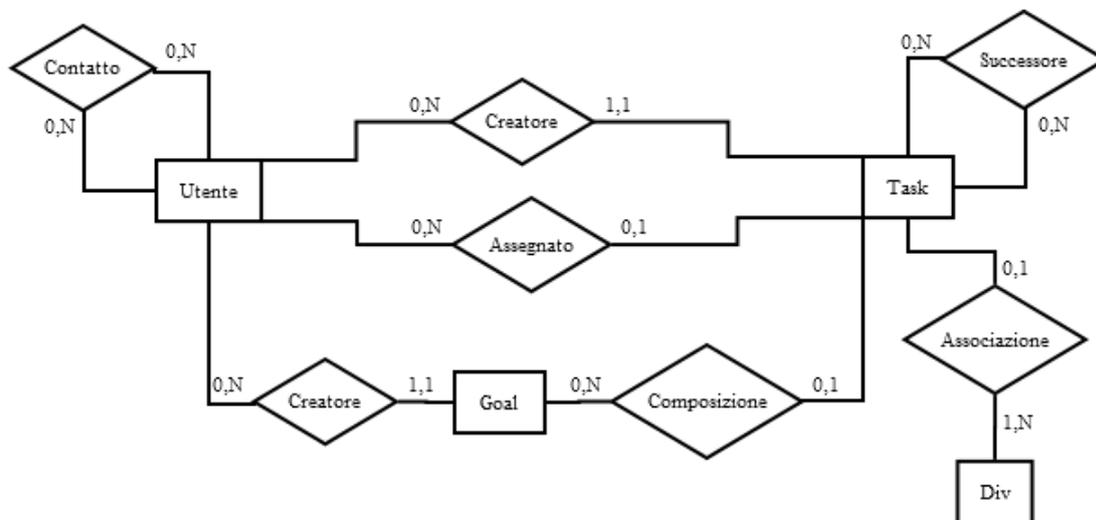


Figura 4.7 – Modello concettuale

Nel modello in Figura 4.7 sono stati omessi gli attributi per mantenere una certa leggibilità dello schema e rimandiamo alle sezioni successive per una descrizione esaustiva; precisiamo solamente che, mentre le entità *Task*, *User* e *Goal* hanno una chiave primaria interna, l'entità *Div* ha come una chiave primaria esterna data dal suo divID e dal taskID del task con il quale è associato e che la relazione *Composizione* possiede un attributo startGoal ad indicare quale task è definito come iniziale in un goal.

Modello logico-relazionale

Definito lo schema concettuale, è necessario a questo punto operare una traduzione verso un modello relazionale dato che abbiamo scelto di utilizzare un database relazionale quale MySQL. Il risultato di tale operazione è visualizzabile nella successiva Figura 4.8.

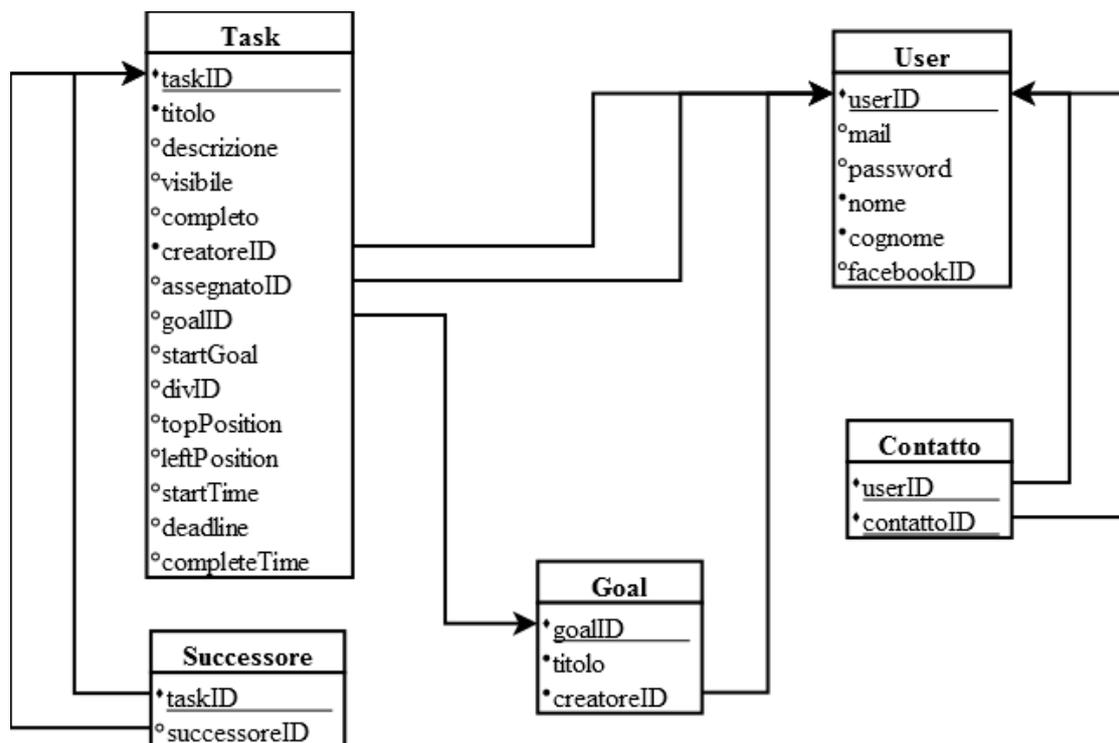


Figura 4.8 – Modello logico-relazionale

Dallo schema presente in quest'immagine possiamo dedurre le cinque tabelle che andranno a comporre il nostro database e i rispettivi vincoli di integrità referenziale dovuti alla traduzione delle varie relazioni del modello concettuale.

Da notare che l'entità *Div* è stata inglobata nella tabella *Task* dato che si tratta di una relazione uno a molti così come quasi tutte le altre relazioni presenti, per questo non è stato necessario definire nuove tabelle; questo però non è stato possibile con le relazioni *Contatto* e *Successore* trattandosi di relazioni molti a molti.

Descrizione tabelle e attributi

Il database, abbiamo visto, è complessivamente composto da cinque tabelle *User*, *Task*, *Goal*, *Contatto*, *Successore*. La prima che analizziamo (*User*) è quella relativa agli utenti i cui attributi sono i seguenti:

- userID – è il codice univoco di un utente indispensabile per il riconoscimento da parte del sistema;
- mail – indirizzo e-mail al quale associare in maniera univoca il profilo degli utenti;
- password – insieme all'indirizzo e-mail costituisce la coppia di credenziali per il login al sistema;
- nome – nome dell'utente;
- cognome – cognome dell'utente;
- facebookID – se l'utente effettua il login tramite il popolare social network, in questo campo è salvato l'ID che gli è stato associato da Facebook.

Tra gli utenti abbiamo visto che esistono delle relazioni e questi legami sono mantenuti dalla tabella *Contatto*:

- userID – è il codice ID dell'utente;
- contattoID – è il codice ID dell'altro utente con il quale è in contatto.

Le informazioni relative agli obiettivi sono invece salvate nella tabella *Goal*; notiamo comunque che qui sono presenti solo le informazioni base perché per quanto riguarda la struttura e i task che la compongono è logicamente più corretto mantenere i due concetti task e processo separati e poi nel primo avere un collegamento col secondo con una chiave esterna. Gli attributi di *Goal* sono quindi:

- goalID – identificatore univoco per un goal;
- titolo – stringa rappresentate il titolo dell'obiettivo;
- creatoreID – corrisponde all'ID dell'utente che ha creato il goal.

Passiamo ad esaminare le informazioni salvate relative ai task contenute nella tabella *Task*:

- taskID – identificatore univoco nel sistema per un task;
- titolo - stringa rappresentate il titolo del task;
- descrizione – eventuale descrizione per fornire ulteriori dettagli relativi all'attività del task stesso;
- visibile – valore booleano che indica se il task è visibile oppure no;
- completo – valore booleano che indica se il task è stato completato oppure no;
- creatoreID – corrisponde all'ID dell'utente che ha creato il task.
- assegnatoID – corrisponde all'ID dell'utente al quale è stato assegnato il task; se il valore è pari a zero, indica un task personale cioè appartenente all'utente stesso che lo ha creato;
- goalID – corrisponde all'ID del goal al quale appartiene; se il valore è pari a zero, allora il task non è legato a nessun goal ed è quindi indipendente;
- startGoal – valore booleano che indica se il task corrisponde all'attività iniziale di un obiettivo;
- divID – valore che identifica l'ID del riquadro utilizzato per rappresentarlo graficamente nella struttura del processo; ad esso sono associati anche i due prossimi attributi;
- topPosition – valore che definisce la coordinata Y per la rappresentazione del riquadro associato al task;

- leftPosition – valore che definisce la coordinata X per la rappresentazione del riquadro associato al task;
- startTime – indica l'istante preciso in cui il task passa nello stato visibile;
- deadline – identifica l'indice di un array nel quale sono contenute le varie possibili deadline assegnabili ai task;
- completeTime – indica l'istante preciso in cui il task passa nello stato completo.

L'ultima tabella (*Successore*) contiene le informazioni relative ai task appartenenti a qualche processo e rappresenta le liste di adiacenza di ognuno di essi:

- taskID – identificatore di un task;
- successoreID – identifica un task tramite il suo ID con il quale esiste un legame di successione in qualche processo.

Aggiungiamo alcune note riguardanti le informazioni salvate nel database legate al corretto funzionamento dell'applicazione. Innanzitutto l'indirizzo e-mail deve essere univoco per ogni utente ma poiché nella tabella esso può assumere anche dei valori nulli in quanto non è permesso accedere a tale informazione se l'utente effettua il login con le credenziali del social network non è stato possibile definire il vincolo di unicità messo a disposizione dal DBMS: questo controllo viene quindi effettuato manualmente in fase di registrazione di un nuovo utente. Inoltre, per evitare confusioni all'utente, abbiamo impedito la creazione di più goal con lo stesso titolo: in altre parole, dal punto di vista del database, la coppia di valori titolo-creatoreID nella tabella *Goal* costituisce una chiave. Infine per quanto riguarda i contatti è stato adottato un piccolo accorgimento tecnico per la gestione delle relazioni: poiché queste devono essere bidirezionali, cioè se un utente è in contatto con un altro utente allora deve verificarsi anche il contrario, l'instaurazione di una relazione tra due utenti comporta nella tabella *Contatto* una doppia scrittura nella quale la seconda ha i valori da inserire invertiti rispetto alla prima.

4.4.3. Diagramma delle classi

Presentiamo ora il diagramma delle classi dell'applicazione.

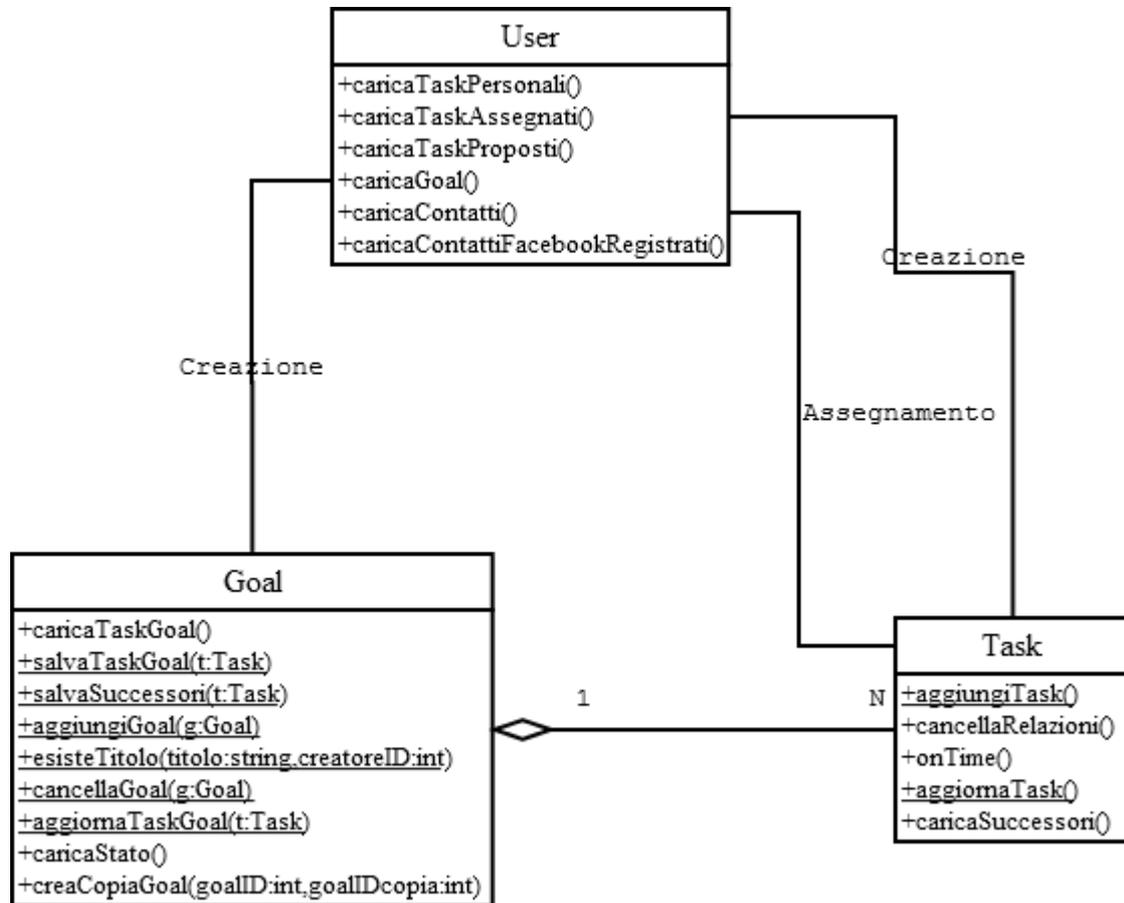


Figura 4.9 – Diagramma delle classi

In Figura 4.9 possiamo osservare le classi realizzate per gestire l'applicazione; nell'immagine abbiamo però ommesso di scrivere tutti gli attributi e i relativi metodi *get* e *set* perché corrispondono esattamente agli attributi delle tabelle del database. Esamineremo solamente i metodi che si rivelano utili ai fini pratici dell'implementazione software.

Classe User

La classe *User* è quella attraverso la quale vengono gestiti gli utenti e le attività ad essi collegati; i metodi riportati sono richiamati non appena un utente effettua il login e il sistema lo reindirizza alla propria home page:

- *caricaTaskPersonali* – recupera i task personali, cioè quelli che sono stati creati e assegnati all'utente stesso;
- *caricaTaskProposti* – recupera i task creati e assegnati a qualche contatto, per osservare in quale stato essi si trovano;
- *caricaTaskAssegnati* – recupera i task che altri utenti gli hanno assegnato;
- *caricaGoal* – recupera gli obiettivi definiti dall'utente;
- *caricaContatti* – recupera i contatti dell'utente;

Il metodo *caricaContattiFacebookRegistrati* serve, invece, a ricavare una lista dei contatti dell'utente che hanno effettuato l'accesso con Facebook: è utilizzato per visualizzare i contatti provenienti dal social network che non sono ancora registrati nel sistema ma questa operazione avviene solo su specifica richiesta dell'utente.

Classe Task

Esaminando la classe *Task*, notiamo la presenza di due metodi statici *aggiungiTask* e *aggiornaTask* che sono richiamati rispettivamente ogni qual volta l'utente crea un nuovo task oppure ne modifica uno esistente: per modifica non si intende solamente la variazione del titolo o della descrizione ma si riferisce anche all'operazione di completamento. Gli ulteriori tre metodi si occupano di:

- *cancellaRelazioni* – eliminare tutti le relazioni di successione ed è richiamato se viene eliminato un intero processo o viene alterata la struttura eliminandone qualche task;
- *caricaSuccessori* – recuperare tutti i successori quando si accede alla struttura di un processo;

- *onTime* – verificare se l'operazione di completamento dell'attività è avvenuta in tempo rispetto alla deadline impostata.

Classe Goal

L'ultima classe *Goal*, ed anche la più importante, fornisce i metodi per la gestione dei processi. Innanzitutto esaminiamo i tre metodi base per definire ed eliminare i goal:

- *aggiungiGoal* – si occupa di creare un nuovo goal con una struttura inizialmente vuota;
- *esisteTitolo* – controlla, in fase di creazione, se l'utente ha già definito un goal con lo stesso titolo;
- *cancellaGoal* – elimina un goal, la sua struttura, i task che lo compongono e le relazioni di successioni salvate sul database.

Oltre questi metodi base, ne sono presenti ulteriori che garantiscono il corretto funzionamento della logica sottostante i processi come descritto nel capitolo precedente:

- *caricaTaskGoal* – recupera dal database i task che compongono il processo e le relazioni di successione esistenti;
- *caricaStato* – fornisce informazioni sullo stato globale del processo, cioè se è completo piuttosto che senza una struttura definita piuttosto che in attesa del completamento di determinati task;
- *creaCopiaGoal* – crea una copia del processo se si tenta una sua modifica mentre la sua esecuzione è già avviata;
- *salvaTaskGoal* e *salvaSuccessori* – registrano tutte le informazioni per memorizzare sul database la struttura del processo;
- *aggiornaTaskGoal* – aggiorna lo stato del processo al completamento di un task, rendendo visibili eventuali task in attesa di essere sbloccati.

Attraverso questi metodi il sistema può soddisfare tutte le richieste che provengono dall'utente e restituire i risultati desiderati. Nella sezione seguente esamineremo le tecnologie utilizzate e alcuni dettagli implementativi adottati durante la fase di sviluppo dell'applicazione.

4.5. Tecnologie utilizzate e dettagli implementativi

4.5.1. Linguaggi di sviluppo

Per lo sviluppo dell'applicazione lato server si è fatto ampio uso del linguaggio di programmazione PHP, versione 5.3, che offre un supporto molto migliore per uno sviluppo object-oriented. Tutto il codice delle classi è stato scritto utilizzando questa tecnologia così come anche alcune pagine dinamiche che hanno il compito di rispondere adeguatamente alle richieste pervenute dagli utenti tramite chiamate asincrone (AJAX).

Per lo sviluppo lato client si è fatto largo uso di HTML, CSS e Javascript: principalmente i primi due linguaggi citati sono stati utilizzati per la parte statica dell'interfaccia grafica (layout, menù, tabelle, form) mentre Javascript è stato destinato alla parte dinamica, cuore dell'applicazione lato client, quale la definizione dei processi. La scelta si è orientata verso i nuovi standard, HTML5 e CSS3, che forniscono nuovi potenti strumenti per gli sviluppatori che semplificano procedure che fino a poco tempo fa richiedevano uno sforzo per gli sviluppatori: nell'applicativo si possono osservare le nuove specifiche di HTML5 all'opera nei campi delle form che adesso sono tipizzati e non c'è più il bisogno di inserire i controlli per la verifica della correttezza dei dati inseriti dall'utente, ad esempio ora possiamo definire un campo e-mail e avere già il controllo sul formato del testo inserito, oppure possiamo osservare le capacità offerte della tecnica del drag&drop nell'assegnamento degli utenti e nella definizione dei processi o l'uso del nuovo elemento canvas che permette di disegnare con poche righe di codice (nel nostro caso è stato utilizzato per il disegno dei collegamenti nel processo).

Con Javascript, invece, riusciamo a gestire alla perfezione tutte le parti dinamiche dell'applicazione ed è indispensabile se vogliamo sfruttare le potenzialità offerte dalla tecnologia AJAX, cioè chiamate asincrone verso il server con le quali possiamo aggiornare anche solamente porzioni di pagina riducendo quindi il traffico dei dati trasmessi e velocizzando le operazioni: l'aggiunta di task, creazione di goal, salvataggio di processi e completamento di attività sono tutte operazioni implementate sfruttando questa tecnologia. Inoltre tutta la parte relativa alla definizione dei processi è stata sviluppata in Javascript: questo ha permesso di annullare le comunicazioni col server fino al momento in cui non si desidera salvare la struttura ma, essendo completamente un linguaggio lato-client, potrebbe leggermente rallentare l'esecuzione su calcolatori non più moderni.

Se da una parte l'utilizzo di queste moderne tecnologie ha portato molti benefici in fase di implementazione, da un'altra abbiamo dovuto far fronte ad una serie di problemi dovuti alla poca diffusione di questi standard: non tutti i browser ancora supportano tutte le specifiche definite dal consorzio W3C e quindi in alcuni casi non è garantito il completo funzionamento della soluzione. Ne è un esempio il browser Opera che ancora non incorpora le funzionalità legate al drag&drop rendendo quindi l'applicazione inutilizzabile almeno per la definizione dei processi.

4.5.2. Dettagli implementativi

Per rendere gradevole l'interfaccia gradevole agli occhi dell'utente è stato sviluppato un template semplice ma adatto agli scopi dell'applicazione; inoltre, come vedremo nella prossima sezione, abbiamo cercato di utilizzare simboli grafici abbastanza intuitivi anche per gli utenti poco esperti nella fase di definizione della struttura di un processo: il riquadro col simbolo "+" per l'aggiunta dei successori, il cestino per l'eliminazione; l'uso consistente del drag&drop in questa fase per spostare box, assegnare utenti e deadline è giustificata dal fatto di voler dare completa libertà all'utente in fase di modellazione: in particolare per correggere un assegnamento errato ad esempio di un task ad un utente è sufficiente trascinare nel suo box il nuovo utente e il sistema provvederà al resto. Altro particolare grafico è la marcatura dei riquadri dei task di un processo: un bordo spesso indica il task iniziale mentre un

bordo doppio sta ad indicare il task o i task finali del processo; inoltre sono stati usati colori per identificare il completamento dei task e se questo sia avvenuto in tempo rispetto alla deadline assegnata: verde per i task completati entro il tempo limite, rosso altrimenti.

Abbiamo già parlato, poi, del fatto di aver integrato solamente un social network (Facebook) all'interno dell'applicazione ma ciò non vieta di poterne utilizzare qualsiasi altro: la scelta è ricaduta su Facebook data la sua enorme popolarità e il fatto di dare la possibilità di postare messaggi sulle bacheche degli utenti e di gestire la loro rete di amici senza particolari complicazioni grazie al supporto fornito dalle sue API. In realtà esistono molte versioni di queste librerie, quasi una per ogni linguaggio di programmazione web esistente, ma la nostra scelta si è diretta verso la libreria PHP in accordo con il linguaggio utilizzato lato-server; inoltre è presente anche la versione Javascript per un'integrazione lato-client ma era una versione ancora troppo acerba in quanto completamente rinnovata qualche mese fa e, quindi, poco documentata e testata.

5. Esempi d'uso

5.1. Aspetti generali

Presentiamo in questa sezione due esempi che riteniamo abbastanza esplicativi per illustrare il comportamento reale della soluzione implementata: nel primo studieremo la gestione di un processo senza utilizzare le funzionalità offerte dall'integrazione con il social network per concentrarci sugli aspetti comportamentali del sistema in risposta ai cambiamenti che avvengono durante l'esecuzione dell'intero processo mentre nel secondo definiremo un goal più semplice con il quale, però, evidenzieremo gli aspetti social.

Prima di affrontare queste analisi, esaminiamo brevemente gli aspetti più generali dell'applicazione quali la gestione di task singoli e dei contatti. In Figura 5.1 mostriamo la home page che si presenta all'utente dopo aver superato la fase di autenticazione.

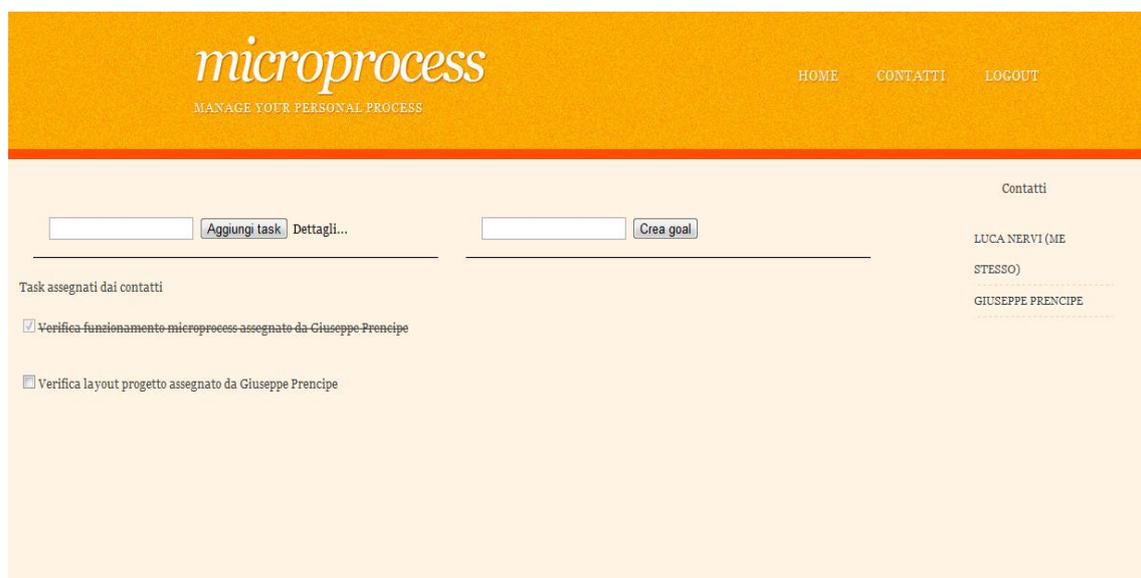


Figura 5.1 – Home page utente

La parte centrale dell'interfaccia è suddivisa in due colonne affiancate dove, nella prima, compariranno i task personali, quelli assegnati ai propri contatti, con i quali non può interagire, e quelli che invece i contatti gli hanno assegnato mentre la seconda è dedicata all'elenco degli obiettivi creati e al loro stato. Di lato si possono osservare l'elenco dei propri contatti ai quali è possibile assegnare dei task con il semplice drag&drop.

Nell'immagine successiva (Figura 5.2) mostriamo la pagina per la gestione dei contatti raggiungibile attraverso il link presente nel menù in alto, posizionato affianco al titolo dell'applicazione.

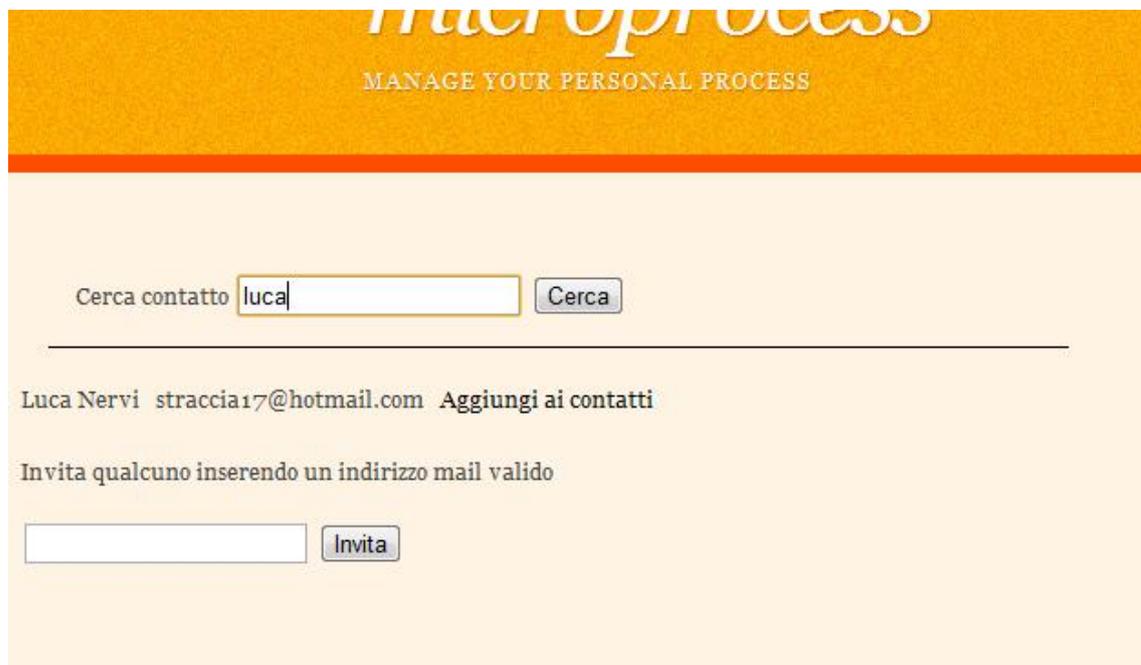


Figura 5.2 – Pagina contatti

In questa pagina è possibile fare ricerche sugli utenti già registrati nel sistema ed è poi possibile aggiungere i risultati tra i propri contatti; nel caso in cui la persona cercata non sia ancora registrata è possibile mandarle un invito alla sua casella e-mail inserendo il suo indirizzo nel campo di testo. Nel caso l'utente si fosse loggato nel sistema utilizzando le credenziali d'accesso del social network, in questa finestra

compare anche un comando per caricare i contatti del profilo social e invitarli tramite un messaggio pubblicato sulla bacheca del contatto.

5.2. Esempio – Gestione processo

Questo esempio illustra il comportamento del sistema durante la gestione di un processo, dalla fase di definizione fino al completamento dell'esecuzione.



Figura 5.3 – Creazione processo

Il primo per la definizione di un processo è la creazione del goal associato attraverso la form; quindi questo comparirà nella lista sottostante con i rispettivi link per visualizzarne i dettagli oppure eliminarlo e conterrà anche una stringa che indicherà lo stato complessivo del processo. Proseguendo verso la pagina dei dettagli si apre la pagina che permette la costruzione della struttura del processo (Figura 5.4).

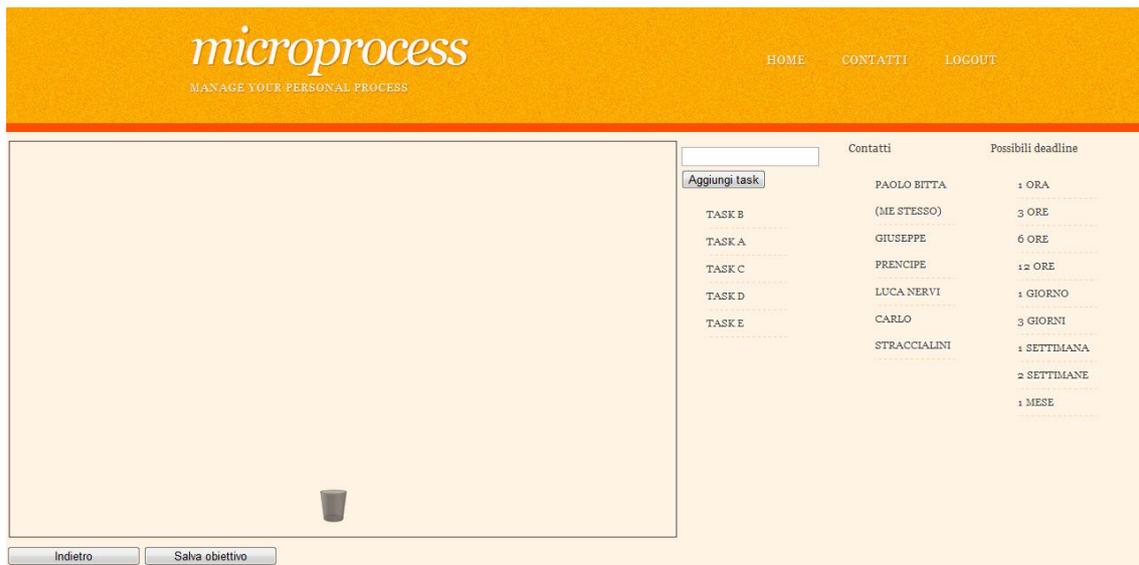


Figura 5.4 – Dettagli goal

L'immagine mostra un pannello dove trascinare i task del goal, che devono essere creati prima di poter agire su di essi e precisiamo, inoltre, che non esiste nessun legame con quelli che l'utente può creare nella propria home page che sono e rimangono indipendenti da questo contesto, per disegnare lo schema del processo. Il primo task trascinato all'interno del riquadro sarà impostato automaticamente come task iniziale del processo, tutti i task successivi dovranno essere legati in qualche modo a questo task altrimenti il sistema ne impedirà l'aggiunta.

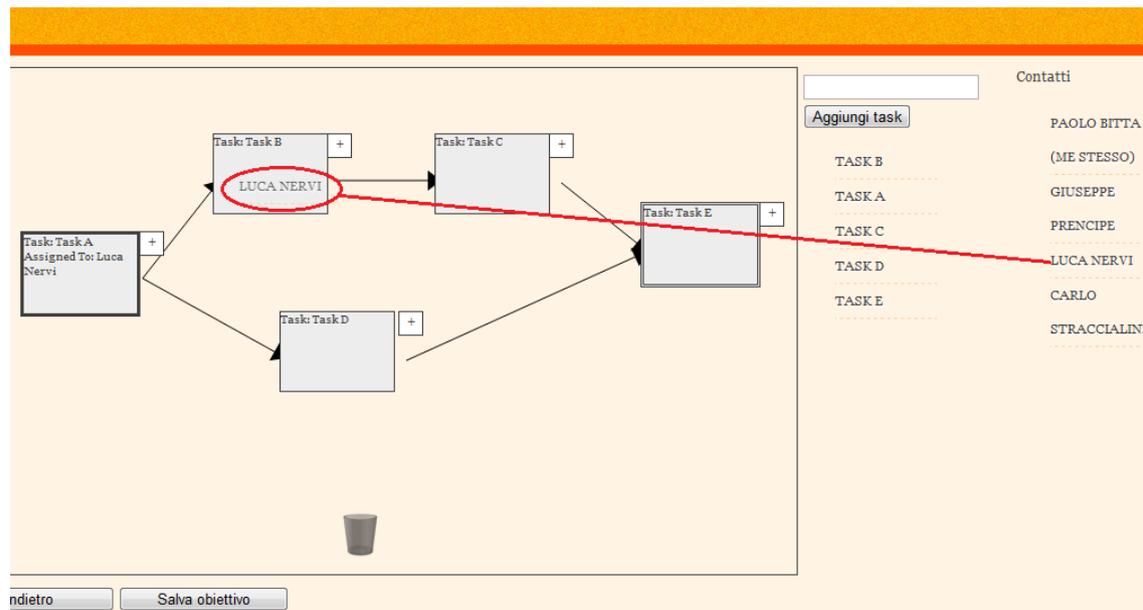


Figura 5.5 – Creazione processo

Nella Figura 5.5 possiamo osservare la struttura completa del processo nella quale sono messi in risalto il task iniziale con un bordo più marcato rispetto agli altri task e quello finale con un bordo doppio; con la solita tecnica del drag&drop poi si impostano le deadline e si assegnano i task agli utenti. Al termine delle operazioni si procede con il salvataggio della struttura dell'obiettivo. A questo punto il sistema gestirà le operazioni relative ai task: inizialmente solo il task iniziale sarà visibile come si può osservare dalle seguenti due immagini.



Figura 5.6a – Situazione iniziale processo: vista creatore

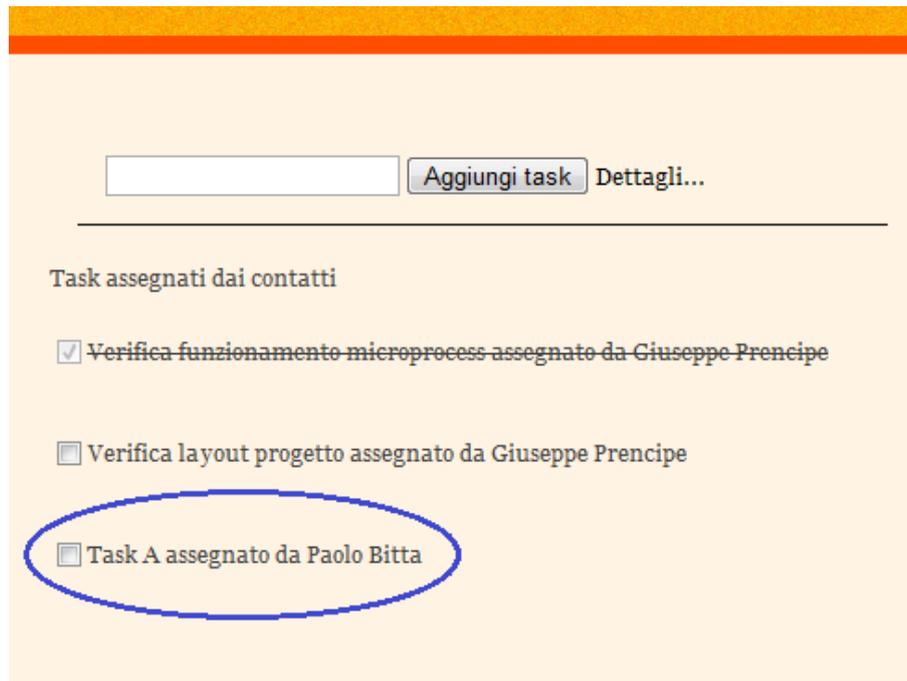


Figura 5.6b – Situazione iniziale processo: vista utente

In Figura 5.6a possiamo osservare la situazione dal punto di vista del creatore del processo: nella colonna di destra notiamo il task iniziale del processo assegnato all'utente Luca Nervi che ancora non è stato eseguito così come riportato anche nella colonna di sinistra dalla stringa che informa sullo stato complessivo dell'esecuzione del processo. Dal punto di vista dell'utente Luca Nervi (Figura 5.6b) troviamo un task che ci è stato assegnato e che dovremo in qualche modo portare a termine; fino a che ciò non accadrà il processo rimarrà in una fase di stallo.

Nell'immagine successiva mostriamo la situazione che si incontra non appena il task iniziale è stato completato.



Figura 5.7 – Primo task completato

Data la struttura mostrata in Figura 5.5, l'esecuzione del task iniziale del processo comporta l'abilitazione dei suoi due task successivi che sono assegnati a due utenti distinti: in questa maniera possiamo sfruttare il parallelismo delle attività ed accorciare, quindi, i tempi di esecuzione totali del processo. In Figura 5.7 possiamo osservare il task iniziale marcato come completo, la comparsa dei due nuovi task e lo stato del goal che ci conferma la stessa situazione.

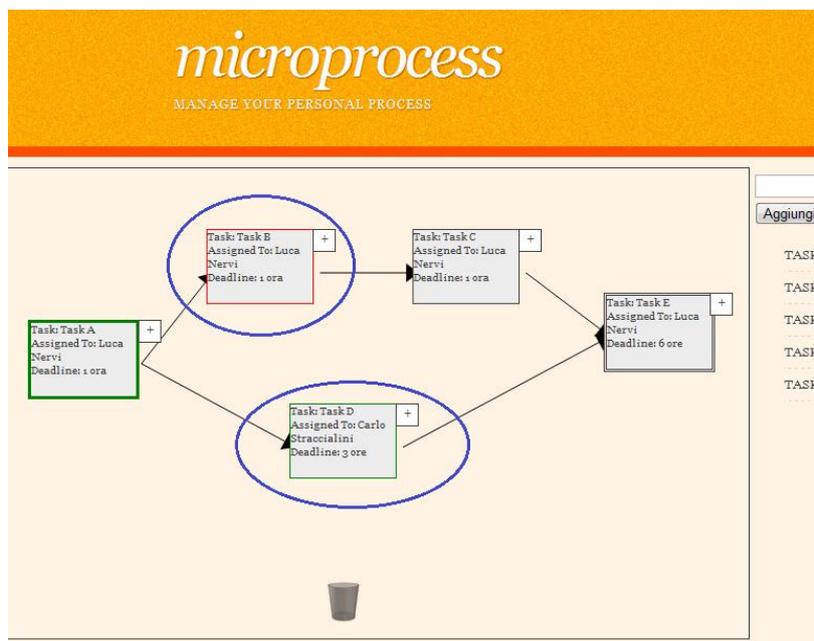


Figura 5.8 – Task e deadline

Nell'immagine in Figura 5.8 possiamo osservare la distinta colorazione del bordo del riquadro che contraddistingue il fatto che un task sia stato completato entro la deadline prefissata oppure abbia oltrepassato i limiti di tempo imposti: nel primo caso il bordo è colorato di verde come si può vedere per il task iniziale e per il Task D mentre nel secondo caso è stato utilizzato il colore rosso ad indicare l'avvenuto ritardo nell'esecuzione.

Come possiamo vedere dall'immagine successiva (Figura 5.9), in questa situazione solo un task è nello stato è stato reso visibile cioè il Task C assegnato all'utente Luca Nervi; l'ultimo task del processo, il Task E, invece risulta ancora bloccato nel suo stato perché non tutti i suoi predecessori sono stati completati cioè non è ancora rispettata la condizione affinché diventi visibile. Solo con il completamento del Task C, l'ultimo task del processo diventerà visibile e potrà essere eseguito.



Figura 5.9 – In attesa del Task C per completare l'ultimo task

Eccoci infine alla conclusione del processo come riportato dalle Figure 5.10a e 5.10b.



Figura 5.10a – Obiettivo completato

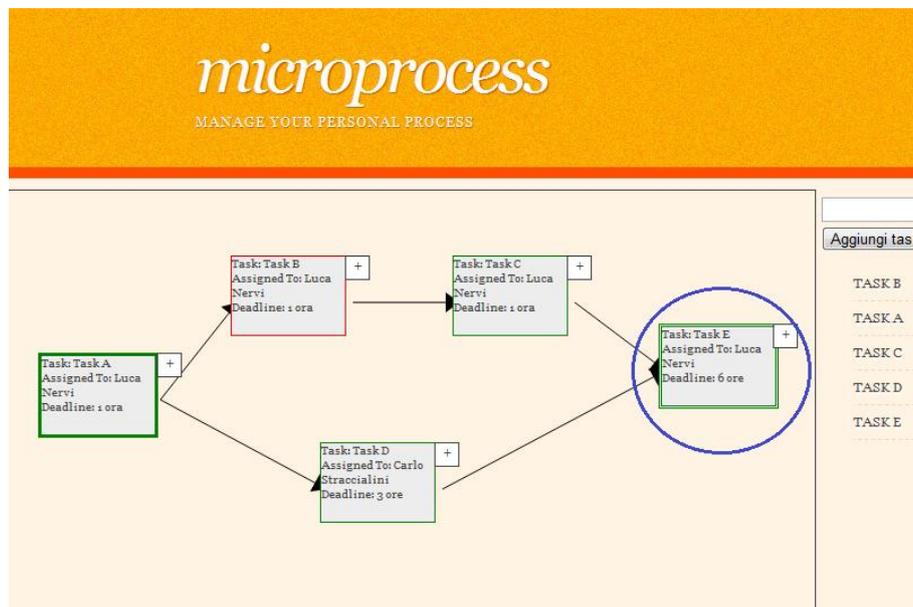


Figura 5.10b – Dettagli obiettivo completato

Nella prima immagine (Figura 5.10a) osserviamo il sistema che ci notifica il completamento dell'obiettivo nella stringa indicante lo stato del processo mentre la seconda immagine (Figura 5.10b) riporta i dettagli del processo: come si può vedere

tutti i task sono stati completati entro i tempi imposti tranne il Task B, il cui completamento è avvenuto oltre la deadline prefissata.

5.3. Esempio – Integrazione social network

Mostriamo ora un secondo esempio nel quale mettiamo in luce le caratteristiche derivanti dall'integrazione con il social network Facebook con un piccolo processo costituito da due soli task sequenziali, la cui esecuzione è affidata ad un nostro contatto di Facebook non ancora utilizzatore del sistema.



Figura 5.11 – Login con Facebook

Innanzitutto per sfruttare queste caratteristiche è necessario che l'utente effettui il login attraverso Facebook; per poterlo fare basta semplicemente cliccare sul link cerchiato in Figura 5.11, non serve alcuna registrazione preventiva ma al primo collegamento viene richiesto dal social network di dare il consenso al trattamento di alcuni dati utili. Dopo alcuni secondi avremo davanti agli occhi la solita home page personale e nella colonna dei contatti saranno aggiunti automaticamente i nostri amici di Facebook che già utilizzano la nostra soluzione.

Nell'eventualità possiamo invitare qualsiasi nostro amico tramite la pagina dei contatti dove sarà presente un collegamento per mostrare la lista dei contatti del social

network ai quali possiamo mandare un invito sulla loro bacheca come mostrato in Figura 5.12.



Figura 5.12 – Invito tramite Facebook

A questo punto vediamo come si integrano processi e social network: in Figura 5.13 mostriamo il semplice processo formato da due task, entrambi assegnati all'utente che abbiamo invitato e che nel frattempo ha iniziato ad utilizzare il sistema; se così non fosse stato, non avremmo mai potuto assegnargli dei task da eseguire.

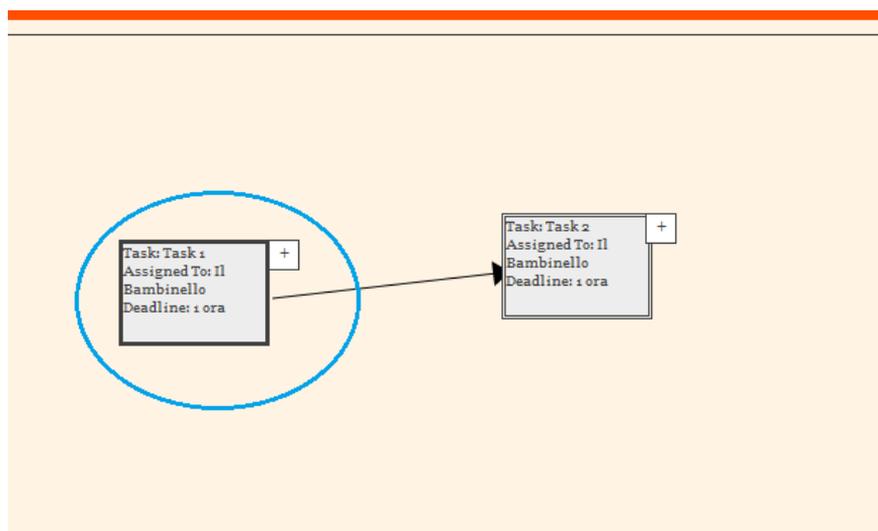


Figura 5.13 - Processo social

Salvando il processo il sistema segnala all'utente che gli è stato assegnato il Task 1 sfruttando il sistema di notifiche del social network e quindi riceverà sulla sua bacheca un avviso come quello mostrato nella successiva Figura 5.14.



Figura 5.14 – Notifica Task 1

L'utente dovrà quindi completare il task e marcarlo come completato nel solito modo come abbiamo già visto nell'esempio precedente; tocca quindi al sistema rendere visibile il task successivo, notificare allo stesso utente l'assegnamento di un secondo task (come definito dal processo) e notificare al creatore del processo l'avvenuto completamento del primo task. Nelle due immagini successive (Figure 5.15a e 5.15b) mostriamo il comportamento del sistema in questa situazione.

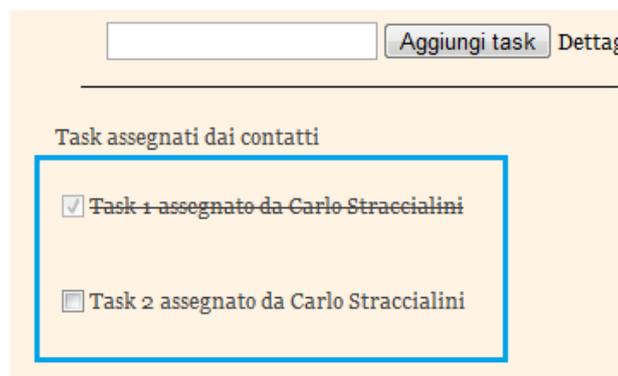


Figura 5.15a – Completamento Task 1: sistema



Figura 5.15b – Completamento Task 1: notifica

Con questi due esempi abbiamo mostrato le caratteristiche più interessanti della nostra soluzione ed illustrato come possono essere utilizzate per supportare le nostre attività quotidiane.

5.4. Valutazioni utenti

Abbiamo chiesto ad amici e conoscenti di testare questa soluzione per alcuni giorni per due motivi principalmente: scoprire eventuali bug o malfunzionamenti e avere un loro feedback sull'uso di questa applicazione.

Per quanto riguarda questo secondo aspetto, possiamo inserire i commenti ricevuti in due categorie: la prima racchiude gli utenti soddisfatti del comportamento della nostra applicazione, della facilità con la quale possono interagire con questo strumento per definire i processi e operare gli opportuni assegnamenti; hanno molto gradito l'integrazione con Facebook che ha permesso loro di essere immediatamente attivi e di non dover perdere tempo a registrarsi e poi autenticarsi e il sistema di notifica dei task implementato. Alla seconda categoria invece appartengono coloro che non sono

riusciti ad utilizzare l'applicazione, o meglio, non ne hanno capito l'utilità e quindi hanno desistito nel proseguire con l'analisi.

Se andiamo ad approfondire i motivi per i quali esiste anche questa seconda categoria di utenti è emerso che:

1. ritengono l'applicazione incomprensibile per una mancata documentazione di supporto e per i pochi feedback che fornisce durante le operazioni;
2. non hanno mai utilizzato simili applicazioni, ne per la gestione dei processi ne per la semplice gestione di una lista di cose da fare, e quindi si sono trovati spaesati;
3. molti di quelli che si sono ritenuti soddisfatti avevano già visto in qualche modo strutture di questo genere o nel loro ambiente lavorativo o durante il periodo scolastico e universitario anche se in contesti completamente differenti alla gestione dei processi.

Possiamo quindi concludere che per abbracciare e soddisfare un pubblico più ampio è necessario fondamentalmente fornire una documentazione di supporto al primo approccio verso l'applicazione e una spiegazione di come questa possa aiutarli ad organizzare alcune attività reali che richiederebbero una maggiore organizzazione in mancanza del supporto informatico. Notiamo però come nessun utente abbia risentito della mancanza di costrutti condizionali e ciclici per la definizione dei propri processi, segno che le semplificazioni apportate sono adatte al pubblico di utenti finali verso il quale ci rivolgiamo.

6. Conclusioni

In questo lavoro abbiamo potuto osservare lo studio e lo sviluppo di un'applicazione per la gestione dei processi personali destinata principalmente ad utenti finali e non ad aziende o imprese, sfruttando anche i vantaggi derivanti da un'integrazione della stessa con i social network. Dopo una prima fase di analisi dei tools già presenti nel web per la gestione delle tasklist personali nella quale abbiamo scoperto che in nessuna di queste soluzioni esiste un approccio di tipo processuale ed aver analizzato, nel frattempo, anche soluzioni appartenenti al mondo enterprise per il BPM e il SocialBPM in particolare, abbiamo cercato di fornire una soluzione formale al nostro problema: siamo passati quindi a modellare i concetti necessari per esprimere in maniera rigorosa il nostro approccio ottenendo una rappresentazione matematica per utenti, task e processi. Per quanto riguarda quest'ultimo punto è stato notevolmente approfondito il discorso definendo le funzioni di assegnamento e di successione per i task e studiando un sistema per la verifica dell'assenza di cicli nel grafo che rappresenta la struttura del processo: la presenza di cicli può, infatti, non garantire la corretta esecuzione del processo e potrebbe condurre ad una situazione di stallo. Inoltre sono stati approfonditi i motivi che ci hanno portato all'eliminazione di alcune classi di costrutti come quello condizionale, cioè la necessità di mantenere l'applicazione ad un livello di apprendimento semplice ed immediato dato il nostro target-utenti, e sono state analizzate meglio quelle caratteristiche dei social network che possono portare benefici alla nostra soluzione. Infine siamo arrivati alla fase di sviluppo e test. Abbiamo seguito i principi dell'ingegneria del software: dapprima è stata effettuata un'analisi approfondita dei requisiti dell'applicazione ed è stata proposta un'idea per un'interfaccia web capace di mostrarsi intuitiva e semplice da utilizzare all'utente finale, si è passati poi alla fase di progettazione durante la quale sono stati definite le strutture per il database e per le classi; per l'implementazione vera e propria, infine, abbiamo fatto largo uso delle ultime tecnologie sviluppate nel campo della programmazione web, specialmente per quanto riguarda l'interfaccia utente con l'utilizzo di HTML5 e CSS3 che con l'inserimento di alcune feature nelle

loro nuove specifiche facilitano il compito degli sviluppatori. Sono stati presentati quindi alcuni esempi per mostrare l'utilizzo dell'applicazione coinvolgendo alcuni utenti per la fase di debugging e di test e ricevere da loro feedback su quali aspetti migliorare e correggere e quali, invece, sono quelli positivi da considerare dei punti di forza della nostra soluzione.

Testando personalmente l'applicazione e ascoltando anche i commenti degli altri utenti, la nostra soluzione porta a compimento gli scopi per i quali è stata progettata: la parte relativa alla gestione dei processi personali opera seguendo il modello che avevamo proposto nel nostro approccio al problema, notifica al momento giusto l'assegnazione dei task agli utenti e il loro completamento e aggiorna ogni volta lo stato nel quale si trova il processo; la modellazione delle strutture tramite la gestione di elementi grafici garantisce immediatezza e intuitività anche se, a detta di chi ha avuto un primo approccio con strumenti di questo genere, avere tra le mani della documentazione di supporto e dei feedback di conferma dell'operazione svolta dall'applicazione potrebbe essere un notevole aiuto per comprendere i meccanismi di definizione dei processi; chi, invece, ha già avuto a che fare con strumenti di modellazione grafica, anche se in settori distinti dalla gestione dei processi, non ha avuto problemi ad adattarsi alla nostra soluzione e ai suoi meccanismi. Per quanto riguarda la gestione singola dei task non si sono verificati problemi ed il loro funzionamento è stato compreso da tutte le categorie di utenti. Un aspetto del quale si può discutere è l'inserimento di un operatore condizionale tra i costrutti utilizzabili nella definizione dei processi perché in alcune situazioni può essere conveniente avere un costrutto di questo genere; ovviamente so dovrà trovare una soluzione che ne permetta anche un uso immediato e senza complicazioni, nell'ottica di mantenere la semplicità d'uso dell'applicazione.

Da questo lavoro possono naturalmente svilupparsi nuovi progetti e prendere corpo alcune idee interessanti:

- nella nostra soluzione è stata proposta solamente l'integrazione con Facebook, ma naturalmente deve essere possibile disporre di più social network in modo tale da espandere i propri rapporti verso nuovi ambienti dato che stanno nascendo sempre più social network specializzati in specifici contesti;
- un'altra idea da sviluppare potrebbe essere il porting dell'applicazione verso piattaforme mobile; di questi tempi si stanno affermando nel mercato molti dispositivi mobile quali tablet e smartphone che dispongono di connessioni in modo da stare sempre online e offrono la potenza necessaria a supportare queste applicazioni grazie alla loro dotazione hardware, quindi pensare di avere sempre con se la gestione delle proprie attività quotidiane dovrebbe essere un'ottima idea;
- un ulteriore sviluppo potrebbe essere quello di fornire dei servizi destinati a supportare le attività definite nei task: ad esempio se si vuole organizzare una partita a calcio, ci sarà un servizio che aiuta l'utente a contattare il più vicino campo disponibile per prenotarlo; si può pensare quindi ad una serie di servizi web per aiutare l'utente nell'esecuzione dei propri task aumentando il livello di automazione dei processi personali.

Bibliografia

- [1]. Stephen A. White – Introduction to BPMN – IBM, Gennaio 2004 -
http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf

- [2]. Jog Roj & Martin Owen – BPMN and Business Process Management -
Popkin Software, Settembre 2003
http://www.omg.org/bpmn/Documents/6AD5D16960.BPMN_and_BPM.pdf

- [3]. Marco Brambilla, Piero Fraternali, Carmen Vaca Ruiz - A model-driven
approach to Social BPM applications - In Social BPM Handbook, BPM and
Workflow Handbook series, Future Strategies, USA, June 2011, pp. 95-112.
[http://www.bpm4people.org/cms/wr-
resource/sbpmHandbookChapter.pdf?attr%3Datt28%26att27%3D3047](http://www.bpm4people.org/cms/wr-resource/sbpmHandbookChapter.pdf?attr%3Datt28%26att27%3D3047)

- [4]. IBM Blueworks Live - <https://www.blueworkslive.com/>

- [5]. BPMgeek – IBM Blueworks Live - [http://bpmgeek.com/blueworks-
live?page=1](http://bpmgeek.com/blueworks-live?page=1)

- [6]. Intalio Social-BPM - <http://www.intalio.com/social-bpm>

- [7]. Michael Rosemann - Personal Process Management -
<http://www.michaelrosemann.com/uncategorized/113/>

- [8]. Ingo Weber, Hye-Young Paik, Boualem Benatallah, Corren Vorwerk,
Liangliang Zheng and Sungwook Kim – Personal Process Management:
Design and Execution for End-Users – Technical report of University of New
South Wales

- [9]. Armin Sander – Personal process management – Armin’s blog -
<http://www.replicator.org/content/personal-process-management>
- [10]. Object Management Group – BPMN 2.0 –
<http://www.omg.org/spec/BPMN/2.0/>

Ringraziamenti

Siamo giunti al termine di questa avventura: inutile dire che è stato un periodo fantastico ma dato che non mi è mai piaciuto guardare al passato dico solo che questo momento è solo un altro inizio e non la fine di qualcosa, almeno ci risparmiamo quel senso di velata tristezza che ognuno di noi prova quando si giunge alla conclusione di un viaggio stupendo; ci riempiamo, invece, di quella voglia di vivere che solo la scoperta di cose nuove ti può dare e non saremo contenti finché non avremo soddisfatto queste nostra voglia.

Ma è doveroso fare alcuni ringraziamenti perché se è stato un periodo fantastico lo si deve solamente a chi ti è stato vicino: prima di tutto alla mia famiglia che è stata sempre vicina in ogni momento, dando sempre qualsiasi tipo di supporto anche andando incontro a dei sacrifici. Poi al professore Marco Brambilla, relatore di questo lavoro, che si è mostrato sempre disponibile e ricco di suggerimenti utili e interessanti. Infine ci sono gli amici: quelli vecchi della triennale ad Ancona che forse sono ancora un po' arrabbiati perché ho scelto di trasferirmi in questa università per completare il mio percorso e li ho abbandonati fisicamente ma non è mancata occasione per rivedersi e non mancheranno sicuramente altre improvvisate; gli amici del mio paese e il gruppo di pazzi aspiranti cantanti, che senza di loro le vacanze e i periodi di svago non sarebbero stati così divertenti; ed infine quelli nuovi, conosciuti da poco, ma che sembrano da sempre nella mia vita.

Se ho dimenticato qualcuno spero non se la prenda, purtroppo l'elenco completo sarebbe stato infinito e chi davvero mi conosce sa che ho già scritto abbastanza per i miei standard!

