

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

FACOLTÀ DI INGEGNERIA DI MODENA

Corso di Laurea in Ingegneria Informatica

**Analisi ed integrazione di un sistema di Workflow
ad una applicazione Web evoluta.**

**Caso di studio : S.U.A.P. per la Pubblica
Amministrazione Locale**

Relatore:

[Chiar.mo] Prof. Sonia Bergamaschi

Tesi di laurea di:

Bertrand Landry Tchuenkam

Correlatore:

Ing. Lorenzo Canali

Anno Accademico 2005/2006

PAROLE CHIAVI

Workflow

business process

Processi aziendali

flusso

1.	<i>INTRODUZIONE</i>	6
1.1	Obiettivo della Tesi	6
1.2	Organizzazione dei capitoli	6
2.	<i>INTRODUZIONE AL WORKFLOW</i>	7
2.1	Definizione del Workflow	7
2.2	Workflow management systems	8
2.3	Business process	9
2.4	L'informazione e la sua relazione con i processi e l'azienda	10
2.5	Software per il workflow disponibile sul mercato	11
3.	<i>GLOSSARIO - TERMINOLOGIE</i>	13
3.1	Introduzione	13
3.2	Definizione dei termini del Workflow	13
3.3	Approfondimento delle terminologie workflow con alcuni esempi	15
3.4	Alcuni termini inerenti alla piattaforma .NET usati in questa tesi	32
4.	<i>K2.NET 2003: PRESENTAZIONE E PROSPETTIVA TECNICA</i>	36
4.1	Presentazione del Prodotto	36
4.2	K2.net 2003 : Caratteristiche generali	37
4.3	Ciclo di vita dei processi d'impresa	38
4.4	Architettura e principali componenti di K2.net 2003	39
4.5	Compatibilità con le norme professionali (BML, Wf-XML e BPEL)	45
4.6	Gestione dei processi distribuiti	46
4.7	Architettura dell' applicazione	47
4.8	Posizione di K2.net 2003 in uno spazio BPM (Business Plan Management)	49
5.	<i>K2.NET 2003 STUDIO : STRUTTURA DI UNA SOLUZIONE WORKFLOW</i>	52
5.1	Introduzione	52
5.2	Struttura generale di una Soluzione	52
5.3	Struttura generale di un Processo Workflow disegnato con K2.net 2003 Studio	53
5.4	Esportazione e test di un Processo	62
6.	<i>K2.NET WORKSPACE E SERVICE MANAGER</i>	66
6.1	K2.net Workspace	66
6.2	K2.net Service Manager	66
7.	<i>ANALISI DELLE REGOLE DELLE ATTIVITÀ DI UN PROCESSO</i>	69
7.1	Introduzione	69

7.2	Le logiche d'impresa(<i>business logic</i>).....	69
7.3	Regole d'Attività(<i>Activity Rule</i>).....	69
7.4	Gestione delle eccezioni e errori.....	73
8.	K2.NET RUNTIME OBJECT MODEL	75
8.1	Introduzione	75
8.2	Definizione	75
8.3	Strumenti forniti dal K2ROM	75
8.4	Gli oggetti di K2ROM (<i>K2ROM Objects</i>).....	76
9.	ANALISI INTEGRAZIONE DI K2.NET CON SOFTWARE ESTERNI	94
9.1	Introduzione	94
9.2	Analisi dell'integrazione.....	94
9.3	Integrazione Dati	97
9.4	Analisi struttura database K2.net.....	99
10.	ANALISI E STRUTTURA DEI DATABASE DELLE APPLICAZIONI ESTERNE A K2.NET 2003	105
10.1	Introduzione	105
10.2	Implementazione della struttura una <i>Solution</i> (Soluzione) workflow di K2.net nel database esterno	106
11.	S.U.A.P. - STRUTTURA DEL MODELLO WORKFLOW	118
11.1	Introduzione	118
11.2	Definizione	118
11.3	Funzionalità	118
11.4	Funzionalità dell'area Back-office del Portale internet	120
11.5	Funzionalità Front-office del Portale internet.....	120
11.6	Diagramma di flusso.....	120
12.	GESTIONE E STRUTTURA DEI DATABASE DI K2 E DEL PORTALE INTERNET SUAP	131
12.1	Gestione database utenti	131
12.2	Problemi riscontrati	131
12.3	Risoluzione dei problemi riscontrati.....	131
12.4	Architettura dei database per la gestione dei processi.....	133
12.5	Le variabili del processo.....	135
12.6	Creazione dei form per la varie attività del Processo dal Portale	137
13.	SIMULAZIONE DEL SISTEMA WOKFLOW SUAP	141
13.1	Introduzione	141
13.2	Gestione degli utenti usati durante il test	141

13.3	Avvio e Gestione del processo	143
14.	<i>CONCLUSIONI</i>	147
15.	<i>BIBLIOGRAFIA</i>	148
16.	<i>APPENDICE</i>	150
16.1	Requisiti software per l'installazione.....	150
16.2	Architetture hardware	150
16.3	Installazione di K2.net 2003	153

1. INTRODUZIONE

1.1 Obiettivo della Tesi

L'obiettivo di questa tesi è lo studio dell'implementazione di un sistema di Workflow, la sua interazione con software e database esterni. Il software di Workflow oggetto dello studio si chiama *K2.net 2003* e viene implementato in una società d'informatica che realizza Portali internet e vari software applicativi per la Pubblica amministrazione ed aziende. Sarà studiato il modo in cui i vari software e database saranno modificati e strutturati per interagire con il sistema Workflow.

1.2 Organizzazione dei capitoli

Prima di procedere con lo studio, nel secondo capitolo c'è una breve introduzione al sistema Workflow. In questo capitolo sono spiegati, con vari esempi pratici, le varie terminologie del sistema Workflow e le sue varie applicazioni, Inoltre sarà elencato una serie di applicazioni Workflow, anche *Open Source*, alternativi a *K2.net 2003*. Nel terzo capitolo ho voluto prima di proseguire spiegare e definire alcuni termini del sistema Workflow. Il capitolo quattro è dedicato alla presentazione del software *K2.net 2003* che sarà usato per la gestione del workflow. L'analisi dei componenti importanti di *K2.net 2003* come *K2.net Studio*, *K2ROM* e il *Service Manager* viene eseguito nei capitoli cinque, sei, sette e otto. Nei capitoli nove e dieci sarà eseguito l'analisi necessaria per l'integrazione del software di *K2.net 2003* con applicazioni esterne. L'analisi, l'implementazione e l'esecuzione pratico del caso di studio [Sportello unico alle Imprese (SUAP)] sono descritti nei capitoli undici, dodici e tredici.

Tutte le informazioni inerenti all'installazione del software *K2.net 2003* sono presenti nell'appendice

2. INTRODUZIONE AL WORKFLOW

“Collaborare bene... produrre meglio”, questo è titolo di un articolo scritto da *Gian Franco Stucchi* che scrive *“Co-branding, co-operation, co-opetition, co-location, co-makership, etc. Da qualche tempo nella letteratura ICT c’è un trionfo del prefisso “co”, usato in tutti contesti per indicare l’alto valore socio-economico del “fare insieme” e la sua superiorità su ogni altro modello comportamentale”*. Questa citazione consente di introdurre il tema della collaborazione, un termine oggetto di molte interpretazioni ma che nel nostro contesto può essere inteso come un processo di creazione condivisa della conoscenza, nell’ambito del quale due o più entità (individui o unità organizzate), in possesso di abilità e competenze complementari, interagiscono per raggiungere un obiettivo. Uno dei settori nei quali le strategie collaborative si stanno affermando con sempre maggiore intensità è certamente quello dei sistemi di produzione nell’azienda o gestione pratiche nella pubblica amministrazione. Gli strumenti che favoriscono la collaborazione possono essere chiamati “sistemi di workflow management”.

2.1 Definizione del Workflow

Un sistema di Workflow offre la possibilità di codificare l’organizzazione del lavoro automatizzandola. Ci sono diverse definizioni del termine workflow ma quello più autorevole è del Workflow Management Coalition (www.wfmc.org). L’organizzazione stabilisce, come definizione di workflow, la seguente *“The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules”* tradotto in italiano diventa: ***“L’automazione di tutto o parte un processo di business, durante il quale documenti, informazioni o attività sono passati da un partecipante all’altro per essere elaborati in accordo ad un insieme di regole procedurali”***.

Ogni fase di lavoro viene attivata da una richiesta iniziale da parte di un utente, e produce come risultato un’informazione da restituire all’utente stesso o da trasferire ad altre fasi. Appositi canali di comunicazione garantiscono i trasferimenti di queste informazioni e ognuna

delle attività individuate può essere eseguita non appena si rendono disponibili i dati necessari per il suo svolgimento.

2.2 Workflow management systems

2.2.1 Definizione

I *Workflow management systems* sono realizzati come veri e propri programmi software (*workflow engine*) da installare sui computer dei collaboratori. Così com'è accaduto per le *intranet* (che condividono con i Workflow management systems il ruolo di intermediazione dei flussi di informazione all'interno delle strutture produttive) anche questi sistemi possono essere progettati senza l'utilizzo di protocolli proprietari. Molti vantaggi del workflow management sono dovuti soprattutto all'utilizzo della rete Internet per il mantenimento e l'organizzazione dei contatti, soprattutto quando il gruppo di lavoro è vasto e disperso nello spazio.

La teoria e le applicazioni del **workflow management** promuovono la gestione dei gruppi di lavoro collaborativi secondo il *workflow model*, modello processuale. Un processo consiste in una o più attività ognuna delle quali rappresenta un lavoro da svolgere per giungere a un obiettivo comune. Il workflow management sostiene l'organizzazione del processo di lavoro mediante l'utilizzo di software specifici. Le attività possono essere svolte dai partecipanti (uomo) o da applicazioni informatiche (macchina). I sistemi software che sanno interpretare una serie di regole procedurali integrano diverse funzioni: utilizzano strumenti dell'*Information technology* per la condivisione dell'informazione e gestiscono la comunicazione e il passaggio di compiti da un collaboratore all'altro sono chiamati *Workflow management systems*.

2.2.2 I vantaggi per le aziende

I vantaggi dei supporti *Workflow management systems* sono così sintetizzati:

- **Incremento dell'efficienza** – l'automazione di molti processi fornisce l'eliminazione dei passi non necessari.
- **Migliore controllo del processo** – mediante la standardizzazione dei metodi di lavoro e la disponibilità di strumenti di verifica.

- **Flessibilità** – il controllo del software sul processo di lavoro può essere programmato in base alle esigenze.
- Migliore qualità di servizio (c'è sempre la risorsa giusta al posto giusto).
- Minori costi d'addestramento del personale (il lavoro è guidato dal sistema).

In tali sistemi risulta centrale il ruolo di presentazione, conservazione e condivisione della conoscenza che nasce dalle forme di workflow collaborativo. Le conoscenze individuali, per essere utili in un gruppo di lavoro, devono essere archiviate nel sistema e contenere metainformazioni che ne permettano il reperimento. Questo aspetto ha a che fare con quel filone di studi chiamato *knowledge management*.

Secondo il Dott. Marcello Sabatini di Mscountring, l'azienda ideale ha trovato un'organizzazione che le permette di trasformarsi per il mantenimento o l'acquisizione del suo vantaggio competitivo. **L'azienda ideale ha introdotto l'innovazione tra i suoi processi di supporto.**

2.3 Business process

Un **processo d'impresa** (*business process*) è un processo che coordina l'organizzazione e la politica dell'impresa allo scopo di raggiungere alcuni degli obiettivi dell'impresa.

Ciclo di vita di un processo d'impresa (*business process*)

Il ciclo di vita di un processo d'impresa (*business process*) non è altro che la sequenza delle operazioni che un programmatore deve compiere per realizzare, monitorare e fare funzionare un processo d'impresa.

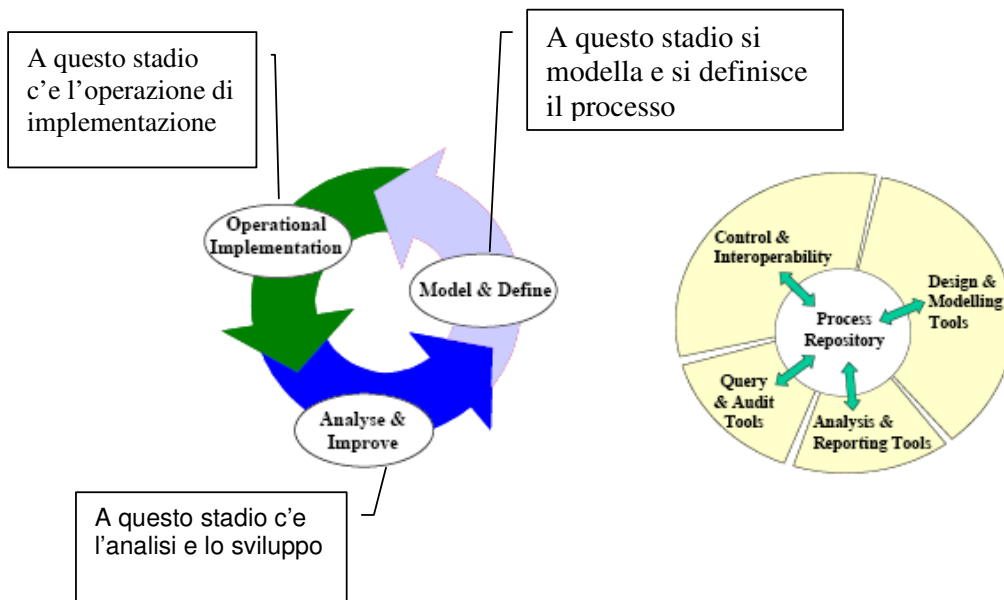


Figura 2.1 :Ciclo di vita di un *business process*

In questo tipo di architettura si vede bene dal grafico di sopra che il *Process Repository* (archivio del Process) è il punto centrale con le interfacce di strumenti e software per interagire con esso durante il ciclo di vita.

2.4 L'informazione e la sua relazione con i processi e l'azienda

Il processo, l'informazione e l'azienda sono inesorabilmente collegati; uno può avere un modello d'architettura da ognuno delle tre dimensioni ma per coerenza devono essere tutte collegate. Il diagramma di sotto mostra le relazioni tra le tre dimensioni.

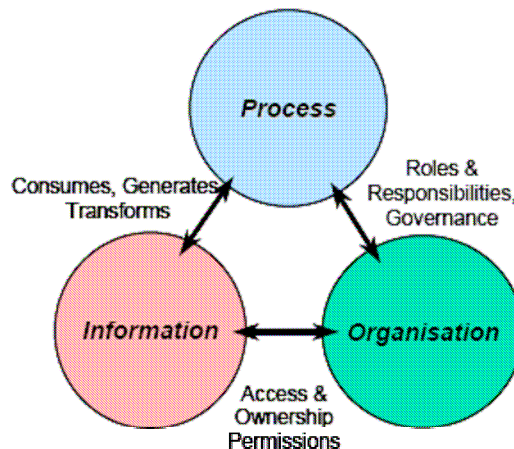


Figura 2.2 : Modello di relazione tra informazione, processi e azienda

2.5 Software per il workflow disponibile sul mercato

Oltre al software K2.net 2003 (che opera in ambiente Microsoft e si appoggia al linguaggio .NET)che è oggetto della mia tesi ci sono altri programmi (anche open source o freeware) che operano in ambiente Unix, Linux ...

Ecco un elenco di software con i vari requisiti di funzionamento

Nome	Requisiti di sistema	Altre informazioni	Licenza	Link download - informazioni
WebAsyst Workflow Management Software 1.1	Win98, ME, NT 4.x, 2000, XP, Unix, Linux	-PHP server -HTML e Windows client - MySQL servers - più database Mysql con licenza Open source - 1 database Mysql con licenza Hosting	Open source: a pagamento Hosting : a pagamento	http://www.brothersoft.com/business/project_management/webasyst_workflow_management_software_download_30588.html http://www.work-tracking.net/
opensymphony		Usa solo dell' xml per la definizione del workflow	Licenza Compatibile con Apache software	http://www.opensymphony.com/osworkflow/
Enhydra Shark	- Apache Tomcat	Linguaggio : -J2EE i processi sono definiti con XPD L	Open source	http://www.enhydra.org/workflow/shark/index.html
Visual paradigm	-window -unix -IBM/OS400	-Visual Studio .net -NetBeans - Jbuilder -Sun	A pagamento	http://www.visual-paradigm.com/product/bpva/
JBoss jBPM	- Apache Tomcat	Linguaggi : - jPDL e BPEL	Open source	http://www.jbpm.org/
WfMOpen		Linguaggio : -J2EE i processi sono definiti con XPD L		http://wfopen.sourceforge.net/
Open Business Engine		Linguaggio : -J2EE	open source	http://www.openbusinessengine.org/index.html

link di altri software di workflow :

<http://www.seagullsoftware.com/products/index.html>

<http://www.handysoft.com/solutions/?src=ADWWKF>

<http://www.cebos.com/Document-Management-System.html?OVRAW=workflow%20software&OVKEY=workflow%20software&OVMTC=standard>

<http://www.intersystems.com/ensemble/>

<http://www.workflowgen.com/>

<http://www.knowledgelake.com/workflow/default.asp?OVRAW=workflow%20software&OVKEY=workflow%20software&OVMTC=standard>

<http://imarkup.com/default.asp?src=overture>

<http://www.mxgraph.com/pages/en/>

3. GLOSSARIO - TERMINOLOGIE

3.1 Introduzione

Questo documento è una traduzione in Italiano del glossario della WorkFlow Management Coalition (WFMC).

Il Glossario della WFMC stabilisce tutti i termini usati nel modello di referenza e gli standard che essa pubblica.

Ho tradotto questa versione di glossario in italiano dal quello francese realizzato da “ Deborah Jouin, DESS de Traduction Spécialisée, Université Grenoble III et François Olléon, MC2

Groupe SAAB Combitech (France)”. Ci sono alcuni termini originali in inglese tradotti in italiano e usati nel seguito della mia tesi solo per rendere più comprensibili alcuni concetti chiave del sistema workflow.

3.2 Definizione dei termini del Workflow

Nella tabella successiva sono definiti i vari termini in inglese usato nel sistema workflow in ordine alfabetico.

INGLESE	ITALIANO
AND-Join	modella la convergenza (Incontro) di due o più percorsi d'esecuzione in un unico percorso: solo quando sono terminati tutti le attività paralleli precedenti , si può eseguire l'attività d'arrivo
AND-Split	Rappresenta la suddivisione del percorso d'esecuzione del processo(A) in sotto-percorsi paralleli(A1, A2, ...): terminata l'attività a iniziale(A) , devono essere eseguite tutte le attività seguenti(A1, A2, ...)
Application Data	Dati applicativi
Audit Trail	Storico
Business Process	Processi d'impresa
Iteration	Iterazione
Manual Process Activity	Attività manuale
Manual Process Definition	Definizione dei processi manuali

Manual Process Execution	Esecuzione dei processi manuali
Manual Process Instance	Caso di processo manuale
OR-Join	modella la convergenza di due o più attività
OR-Split	Rappresenta la separazione dell'esecuzione del processo in due o più percorsi (attività) alternativi, in base a determinate condizioni da porre sugli archi. L'esecuzione potrà percorrere uno solo dei percorsi alternativi.
Organizational Role	Ruolo dell' organizzazione
Parallel Routing	Instradamento parallelo
Process	Processo
Process Activity	attività
Process Activity Instance	Istanza d'attività
Process Definition	Definizione dei processi
Process Definition Mode	Modellazione dei processi
Process Execution	Esecuzione di un processo
Process Instance	Istanza di un processo
Process Role	Ruolo del processo
Route	percorso
Sequential Routing	Instradamento sequenziale
Sub Process Definition	Definizione dei sotto processi
Sub-Process	sotto processo
Tool	Strumenti
Transition Condition	Condizione di transizione
WAPI	WAPI
Work Item	Buono di lavoro
Work Item Pool	Base dei Buoni di lavoro
Workflow	WorkFlow
Workflow Application	Applicazione Workflow
Workflow Enactment	Servizio di Workflow
Workflow Interoperability	Interoperabilità dei Workflow
Workflow Management System	Sistema di gestione del Workflow
Workflow Participant	Attori(partecipante) del Workflow
Workflow Process	Processo Workflow
Workflow Process Activity	Attività del Processo Workflow
Workflow Process Activity Instance	Istanza d'attività del Processo Workflow
Workflow Process Control Data	Dati di controllo del Processo Workflow
Workflow Process Definition	Definizione dei Processo Workflow
Workflow Process Engine	Motore del Workflow
Workflow Process Execution	Esecuzione del Processo Workflow

Workflow Process Instance	Istanza di Processo Workflow
Workflow Process Monitoring	Monitoraggio dei Processi Workflow
Workflow Process Relevant Data	Dati rilevanti per i Processi
Worklist	Elenco (lista) di lavoro
Worklist Handler	Gestione della lista di lavoro

3.3 Approfondimento delle terminologie workflow con alcuni esempi

Le pagine che seguono contengono le definizioni dei termini (secondo la WFCM), con alcuni esempi e dove possibile i vari sinonimi.

Le definizioni sono raggruppate per ogni stadio di sviluppo (Generalità , Esecuzione, Software di Workflow, Amministrazione, Regole, Modellizzazione, Applicazioni e Strumenti Workflow, Dati, Interfaccia e Sviluppo)

3.3.1 Generalità

In questa sezione vengono definiti con esempi i termini fondamentali per un primo approccio al sistema workflow

Processo (Process)

a) Definizione

Un processo è un insieme coordinato di azioni o di operazioni che sono legati, in serie o in parallelo, nello scopo di raggiungere un obiettivo comune

b) Esempio

Processo di atterraggio di un aereo

Processi d'impresa (Business Process)

a) Definizione

Un **processo d'impresa** è un processo che coordina l'organizzazione e la politica dell'impresa nello scopo di raggiungere alcuni degli obiettivi dell'impresa.

b) Esempio

Processo di rimborso spese

Processo di dichiarazione di sinistri (Caso Assicurazioni)

Processo di richiesta credito

⇒ **Studio di un esempio**

Processi d'impresa per chiedere un credito

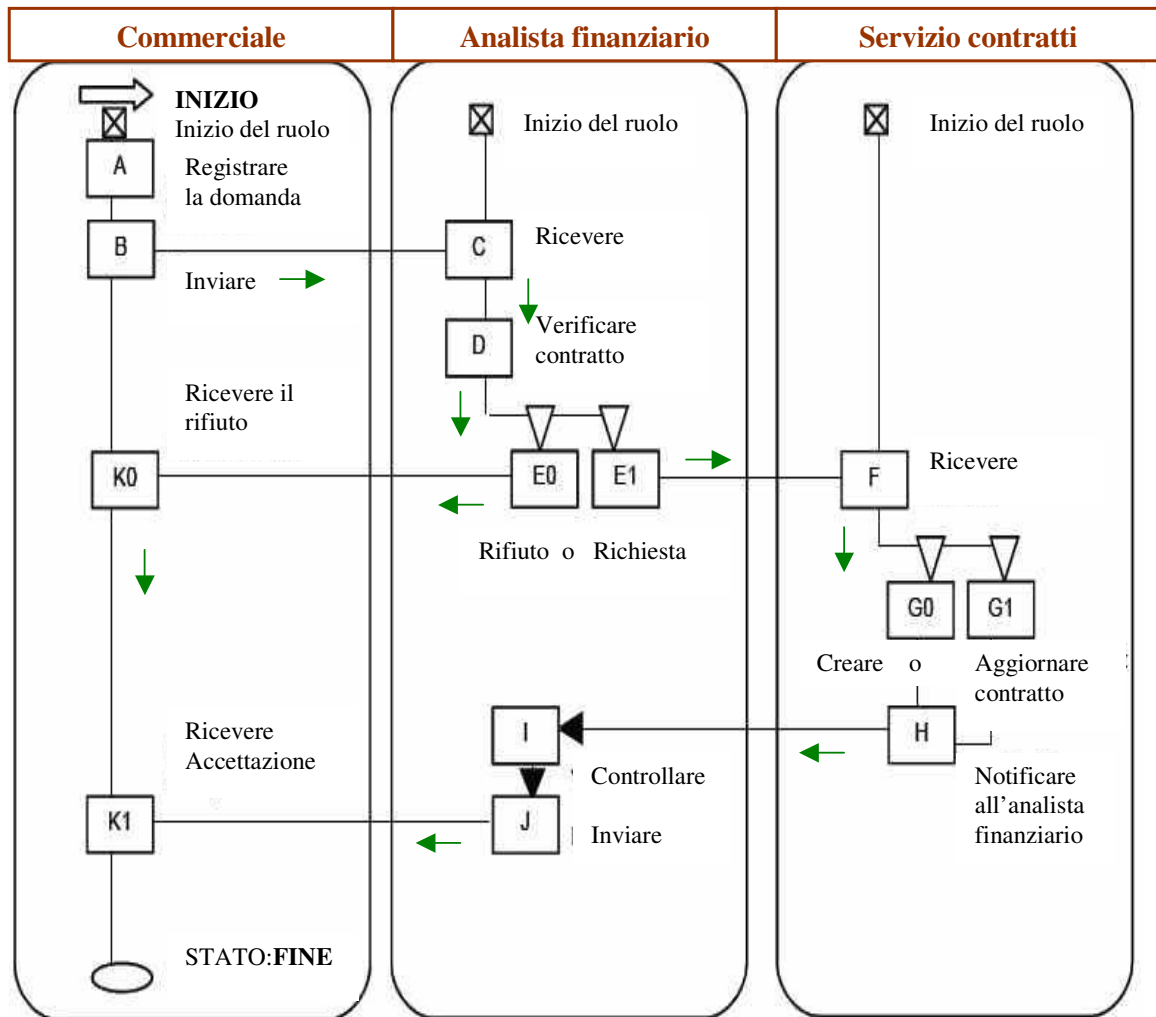


Figura 3.1: Modellizzazione di un processo d'impresa (richiesta credito)

Sono descritti di seguito i vari passaggi :

A- (Il **commerciale** deve) **registrare** la domanda di credito.

- B- (Il **commerciale** deve di seguito) **inviare** questa domanda all'analista finanziario.
- C- (L' **analista finanziario** deve) **ricevere** la domanda.
- D- (L' **analista finanziario** deve di seguito) **verificare** se esiste un contratto.
- E0- (L' **analista finanziario** può) **notificare** al commerciale il rifiuto(immediatamente)...
- E1- (oppure) **chiedere** al servizio contatti il registrare o aggiornare il contratto.
- F- (Il **servizio contatti** deve) **ricevere** la domanda.
- G0- (Il **servizio contatti** può) **creare** un nuovo contratto se non esiste...
- G1- (oppure) **aggiornare** il contratto esistente
- H- (Il **servizio contatti** deve di seguito) notificare all' analista finanziario l'esito del contratto
- I- (L' **analista finanziario** potrà quindi) controllare il contratto ...
- J- (Poi) inviare il suo accordo al commerciale
- K0- (Il **commerciale** deve) **ricevere il rifiuto** ...
- K1- (oppure) **ricevere l'accettazione** del credito

Le attività A a K1 rappresentano un processo d'impresa di una "domanda di credito." Le attività sono attribuite secondo i ruoli, il servizio dei contratti non ha le stesse attività che l'analista finanziario. Queste stesse attività hanno una programmazione logica e l'insieme delle attività ordinate permette di riempire l'obiettivo globale che è di soddisfare(accettare) o respingere la domanda di credito

WorkFlow (*WorkFlow*)

b) Definizione

Ci sono varie definizioni, però cito alcuni di esse :

- ⇒ Un **WorkFlow** è uno strumento informatico dedicato alla gestione dei processi. Questo strumento definisce, gestisce, e esegue dei processi eseguendo dei programmi di cui l'ordine di esecuzione è predefinito
- ⇒ Un **WorkFlow** è l'automazione di tutto o parte un processo di business, durante il quale documenti, informazioni o attività sono passati da un partecipante all'altro per essere elaborati in accordo ad un insieme di regole procedurali. Secondo "Workflow: An introduction – Rob Allen, Open Image Systems Inc., United Kingdom Chair, WfMC External Relations Committee"

b) Uso

È possibile automatizzare alcune parti di un processo usando dei prodotti informatici di Workflow

c) Sinonimi

Gestione di flusso di compiti

Processo WorkFlow (*WorkFlow Process*)

a) Definizione

Un **processo WorkFlow** è un processo di cui l'andamento è controllato da un WorkFlow

b) Uso

Un processo Workflow è in generale composto da molte attività che si collegano allo scopo di costruire modelli di un flusso di lavoro

c) Sinonimi

flusso di lavoro

Sotto-processi (*Sub-Process*)

a) Definizione

Un sotto-processo è un processo che fa parte di un processo più generale. Un processo può così essere composto da, o fare appello, a sotto-processi, di cui alcuni sono manuali e altri sono dei processi WorkFlow

b) Esempio

Il processo di trattamento della posta in entrata (apertura dei pacchi, scelta) è un sotto-processo manuale del processo di trattamento degli ordini ricevuti dalla posta.

Attività (*Process Activity*)

a) Definizione

Un'attività è una tappa di un processo durante la quale un'azione elementare è eseguita.

b) Sinonimi

⇒ Tappa di lavoro o di trattamento o di processo

⇒ Azione

⇒ Operazione

Attività Manuale (*Manual Process Activity*)

a) Definizione

Un'**attività manuale** è un'attività che non è controllata da un Workflow.

b) Esempio

- ⇒ Apertura dei pacchi al servizio posta
- ⇒ Imballaggio di un oggetto da spedire

Attività Workflow (*Workflow Process Activity*)

a) Definizione

Un **attività Workflow** è un'attività che fa parte di un processo Workflow. Consiste nell'automatizzazione di una tappa logico di un processo di cui la realizzazione contribuisce al compimento di un processo Workflow.

b) Esempio

- ⇒ Riconoscenza ottica dei caratteri di un documento digitalizzato
- ⇒ Tappa di convalida di una catena di inserimento dati

c) Uso

Un processo Workflow è composto in generale di una o più attività Workflow che definiscono, una volta connesse, degli itinerari possibili.

3.3.2 Esecuzione

Una volta definito e implementato, il processo può allora essere avviato da una determinata attività. In questa fase il processo è in esecuzione di conseguenza si parla di istanza come elemento di un processo in esecuzione.

Istanza di processo (*Processo Instance*)

a) Definizione

Un'istanza di processo è il seguito delle azioni ed operazioni che sono state effettuate per il trattamento di un caso dato, rispettando il processo da cui dipende questo caso.

Un Istanza di processo può essere uno **istanza di processo Workflow** (*Workflow Process istanza*) o uno **istanza di processo manuale** (*Manual Process istanza*)

b) Esempio

Il trattamento della nota di spese del Sig. Rossi che riguarda il suo spostamento a Roma il 12 gennaio è un'istanza del processo di rimborso di spese di trasferimento.

c) Sinonimi

- ⇒ Caso d'esecuzione
- ⇒ Istanza di processo

Una domanda, una controversia, una nota di spese, una posta, un appello, ecc..., o ogni termine che designa l'oggetto che ha iniziato e che è trattato in un processo.

Istanza d'attività (*Process Activity*)

a) Definizione

Un'istanza d'attività è una tappa di un'istanza di processo. Un'istanza d'attività può essere **un'istanza d'attività manuale (*Manual Activity instance*)**, si tratta allora di un'azione effettuata da un essere umano o una macchina che non è controllata dal Workflow o **istanza d'attività Workflow (*Workflow Activity instance*)**, si tratta allora di un'azione effettuata da un programma automatico o un programma che offre un'interfaccia ad un utente, e che è controllato da Workflow.

Un'istanza d'attività è ad un'istanza di processo ciò che un'attività è a un processo.

b) esempio

Processo : trattamento delle note di spese

Attività : verifica delle note di telefono.

Istanza di processo : trattamento della nota di spese del sig. Rossi per il suo spostamento del 12.2.96

Istanza d'attività : verifica delle note di telefono della nota di spese del sig. Rossi per il suo spostamento del 12.2.96

c) Uso

Un'istanza d'attività può offrire i comandi seguenti:

Armare : l'istanza d'attività può essere avviata

Avviare : l'istanza d'attività comincia il trattamento

Sospendere : sospende l'esecuzione di un'istanza d'attività

Riprendere : Riprendere l'esecuzione di un'istanza d'attività

Visualizzare (ad esempio uno stato) : permette di conoscere lo stato di un'istanza d'attività

Terminare : ferma immediatamente e definitivamente l'esecuzione di una istanza d'attività

Attore del WorkFlow (Workflow Participant)

a) definizione

un attore del WorkFlow è una risorsa (programma automatico, essere umano o essere umano che utilizza un programma avente un'interfaccia utente) che esegue un'attività. Per esteso, è ogni risorsa che effettua parzialmente o completamente il lavoro attribuito ad un'istanza d'attività.

b) Uso

Molti attori del WorkFlow possono esistere per un'attività data. Ad esempio, due persone(attori) possono lavorare in parallelo per effettuare l'attività di digitalizzazione degli elenchi di comando.

Un attore può essere:

- un essere umano
- un programma
- un essere umano che utilizza un programma

Buono di lavoro (*Work Item*)

a) definizione

un buono di lavoro è la rappresentazione del lavoro da effettuare da parte di un attore del WorkFlow nel quadro di un'istanza d'attività.

b) Uso

Un processo è una successione di attività. Un caso di procedura dà luogo ad una successione di istanze di attività. Ogni istanza d'attività è effettuata da un attore del WorkFlow che trova in un buono di lavoro la rappresentazione di ciò che deve fare.

Un attore del WorkFlow può lavorare su uno o più buoni di lavoro nello stesso tempo.

Un attore del WorkFlow può essere in uno degli stati seguenti:

Attivo: Attore del WorkFlow che funziona ed è in fase di esecuzione di almeno un buono di lavoro

Inattivo: Attore del WorkFlow che funziona ma non ha buoni di lavoro da eseguire.

c) Sinonimo

Compito
Unità di lavoro
Elemento di lavoro

Lista di lavoro (*Worklist*)

a) Definizione

Una **lista di lavoro** contiene un elenco di buoni di lavoro da effettuare da parte di un attore del WorkFlow

b) Sinonimi

Elenco di compiti o di buoni di lavoro

Amministratore di lista di lavoro (*Worklist Handler*)

a) Definizione

Un **Amministratore di lista di lavoro** è un programma incaricato di andare a cercare i compiti o buoni di lavoro presso Workflow e di distribuirli nella lista di lavoro (**Worklist**) degli attori del WorkFlow interessati

b) Uso

Un buono di lavoro appare in un canestro. L'utente sceglie un buono di lavoro grazie all'amministratore **di lista di lavoro** e lancia l'applicazione necessaria.

Un amministratore di lista di lavoro può essere un programma a pieno titolo o essere integrato ad un programma avente altre funzioni.

Base dei buoni di lavoro (*Work Item Pool*)

a) Definizione

la base dei buoni di lavoro rappresenta l'insieme di tutti i compiti o buoni di lavoro accessibili.

b) Uso

L'amministratore di lista di lavoro si attende che la base dei buoni di lavoro contenga l'insieme di tutti i buoni di lavoro che si possono ottenere dal Workflow.

c) Sinonimi

Base dei compiti
Raccolta dei compiti (buoni di lavoro)
Serbatoio dei buoni di lavoro (compiti)
Universo dei buoni di lavoro (compiti)

Esecuzione di un processo (*Process Execution*)

a) Definizione

L'esecuzione di un processo designa ciò che si svolge durante il tempo totale tra l'inizio e la fine di un caso di processo.

b) Uso

L'esecuzione di un processo comprende la istanziamento e l'esecuzione di un caso di procedura Workflow, anche l'esecuzione degli eventuali casi di processi manuali associati

c) Sinonimi

Tempo d'esecuzione

3.3.3 Programmi(Software) di WorkFlow

Workflow (*Workflow Process Engine*)

a) Definizione

Un **motore di WorkFlow** è un servizio software che fornisce completamente o parzialmente l'ambiente d'esecuzione di un WorkFlow.

Workflow (*Workflow Enactment Service*)

a) Definizione

Un **servizio di WorkFlow** è un servizio software composto da uno o più motori Workflow dello stesso tipo che servono a definire, gestire ed effettuare processi Workflow.

Sistema di gestione del Workflow (*Workflow Management System*)

a) Definizione

Un sistema di gestione di Workflow è un sistema completo che serve a definire, gestire ed effettuare processi eseguendo programmi il cui ordine d'esecuzione è predefinito in una rappresentazione informatica della logica di questi processi.

b) Uso

Un sistema di gestione di Workflow è composto da uno o più servizi di Workflow.

Un servizio di Workflow è costituito da uno o più motori Workflow dello stesso tipo.

Un sistema di gestione di Workflow può essere ridotto a un solo servizio di Workflow, egli stesso è ridotto a un solo motore di Workflow.

c) Sinonimi

sistema di gestione di flusso di compiti
Workflow

3.3.4 Amministrazione

Definizione dei processi (*Process Definition*)

a) Definizione

una definizione dei processi è una rappresentazione informatica di un processo, che definisce allo stesso tempo i processi manuali e i processi Workflow.

b) Uso

La definizione dei processi può essere creata, modificata ulteriormente, o modificata in occasione dell'esecuzione dei processi.

c) Sinonimi

Modello dei processi
Schema dei processi
Definizione degli smistamenti
Diagramma dei flussi diagramma stato/ transizione
carta dei processi

Definizione dei processi manuali (*Manual Process Definition*)

a) Definizione

La definizione dei **processi manuali** è la parte di una definizione dei processi relativa a quelle che non possono essere automatizzate da un sistema di gestione di Workflow.

Definizione dei processi Workflow (*Workflow Process Definition*)

a) Definizione

La definizione dei **processi Workflow** è la parte di una definizione dei processi relativa a quelle che saranno automatizzate da un sistema di gestione di Workflow.

Definizione dei sotto-processi (*Sub Process Definition*)

a) Definizione

Una definizione di sotto-processo è una rappresentazione informatica di questo sotto-processo.

b) Uso

Una definizione di sotto-processo è utile per definire soltanto una sola volta dei sotto-processi utilizzate in molti processi più generali.

Costruzione di modello dei processi (*Process Definition Mode*)

a) Definizione

La costruzione di modelli dei processi è l'azione che consiste a creare o modificare una definizione dei processi tramite l'attrezzo informatico adeguato.

b) Uso

Il risultato della costruzione di modelli dei processi è la definizione dei processi. Questa definizione dei processi rappresenta la rete delle attività, le loro relazioni ed i criteri che decidono l'inizio e della fine del processo.

c) Sinonimi

Creazione e modifica dei processi
Concezione dei processi
Definizione dei processi (in questo caso il termine rappresenta l'azione ed il risultato dell'azione)

Controllo dei processi (*Workflow Process Monitoring*)

a) Definizione

Il controllo dei processi è l'azione che consiste nel seguire lo svolgimento dei processi Workflow.

b) Uso

Il controllo dei processi è utilizzato dai progettisti dei processi per misurare le prestazioni.

3.3.5 Ruoli

Ruolo organizzativo (*Organizational Role*)

a) Definizione

Un ruolo organizzativo è un elenco di attributi, di competenze e di "know-how" che un attore possiede e mette in pratica. Questo ruolo definisce la posizione dell'attore in un'organizzazione.

b) Uso

Ruolo di supervisore
Ruolo dell'amministratore

Ruolo di processo (*Process Role*)

a) Definition

un ruolo di processo è l'elenco delle attività che un attore può assumere ed effettuare.

b) Uso

Un attore deve assumere un ruolo per potere accedere ai compiti messi a sua disposizione dal sistema di gestione Workflow ed effettuarli.

3.3.6 Costruzione di modelli del Workflow

itinerario (*Route*)

a) Definizione

L'**itinerario** di un caso di processo è il seguito delle attività che sono state attraversate in occasione dell'esecuzione di questo caso preciso. Un processo permette in generale molti itinerari possibili.

Condizione di transizione (*Transition Condition*)

a) Definizione

una **condizione di transizione** è il criterio di progressione, o di cambiamento di stato di un'attività (tappa di lavoro) all'attività (tappa) seguente in occasione di un caso d'esecuzione dato, che si tratti di un processo manuale o automatizzata.

b) Uso

- Esempio: Il processo di dichiarazione di sinistro del cliente X è in transizione tra l'attività valutazione danni e l'attività rimborso perché la condizione di transizione è rispettata.
- Le condizioni di transizione possono essere descritte in occasione della costruzione di modelli dei processi e/o essere calcolato dal sistema di gestione di Workflow durante l'esecuzione dei processi.

Sequenza (*Sequential Routing*)

a) Definizione

Si parla di sequenza quando nel corso di un processo, le attività sono effettuate in sequenza (le une dopo le altre), e che è il solo itinerario possibile.

b) Uso

un buono d'ordine è trattato in una sequenza di tre attività consecutive

Separazione (*OR-Split*)

a) Definizione

Si parla di **separazione** quando un itinerario si apre su molti itinerari possibili e che il caso d'esecuzione segue una o l'altra di quest'itinerari, secondo le condizioni di transizione.

b) Sinonimi

trasmissione, allacciamento, concatenamento, smistamento condizionale
Incrocio

Giunzione (*OR-Join*)

a) Definizione

C'è giunzione quando due o più itinerari convergono verso una stessa attività. Non si tratta della sincronizzazione di molti itinerari, ma piuttosto della giunzione di molti itinerari alternativi.

b) Sinonimi

Convergenza

Allacciamento multiplo (*AND-Split*)

a) Definizione

C'è allacciamento multiplo quando un itinerario unico si separa in due o più itinerari diversi allo scopo di realizzare due o più attività in parallelo.

b) Uso

Può darsi che per un caso di processo dato, ci siano molte istanze d'attività in corso, ad esempio se l'itinerario seguito dal caso di processo comporta un allacciamento multiplo. A seguito di quest'allacciamento, almeno due istanze d'attività possono essere realizzabili

In questo caso, molte liste di lavoro possono contenere ciascuna un buono di lavoro che descrive ciò che occorre fare in ciascuna delle 2 (o più) istanze d'attività in relazione con questo caso di procedura

c) Sinonimi

Esplosione
itinerario multiplo o in parallelo

Percorso parallelo (*Parallel Routing*)

a) Definizione

Si parla di **percorso parallelo** quando su una parte di un processo le attività sono realizzate in parallelo su itinerari multipli.

b) Uso

Esempio : Riempire un formulario

Dati: Il formulario X è composto dalle sezioni A, B e C.

Quando l'attività "riempire un formulario,, è compiuta, le tre sezioni A, B e C sono trattate in parallelo nelle attività corrispondenti,

Tappa trattamento sezione A ,

Tappa trattamento sezione B e

Tappa trattamento sezione C.

Appuntamento (*AND-Join*)

a) Definizione

C'è appuntamento quando due o più attività parallele convergono verso un itinerario unico e si garantisce la sincronizzazione degli itinerari, cioè che si passerà all'attività seguente soltanto quando tutte le attività parallele saranno completate.

b) Sinonimi

Giunzione sincronizzata.

sincronizzazione

Ripetizione (*Iteration*)

a) Definizione

una ripetizione è un ciclo d'attività (s) che implica la ripetizione di una stessa o di molte stesse istanze di attività fino a che una condizione sia soddisfatta.

b) Sinonimi

Ciclo.

3.3.7 Applicazioni e Strumenti WorkFlow

Applicazione Workflow (*Workflow Application*)

a) Definizione

un'applicazione WorkFlow è un programma che contribuisce al trattamento parziale o totale di un buono di lavoro allo scopo di raggiungere l'obiettivo fissato ad un'istanza d'attività Workflow.

b) Uso

Applicazione chiamata dal Workflow: Applicazione avviata automaticamente dal sistema di gestione di Workflow.

. Application Workflow cliente: Applicazione avviata da un attore del Workflow o da un'altra applicazione nel contesto di un buono di lavoro. L'amministratore di lista di lavoro può fare parte di tale applicazione e/o avviare questa applicazione dell'utente.

Strumenti (*Tool*)

a) Definizione

Uno strumento è un'applicazione WorkFlow che di più ha un'interfaccia diretta con/o è chiamato dal sistema di gestione di Workflow tramite l'interfaccia di programmazione d'applicazione del Workflow

3.3.8 Dati

Dati Applicativi (*Application Data*)

a) Definizione

I dati applicabili sono i dati specifici per l'applicazione alla quale il sistema di gestione di Workflow non ha accesso.

b) Uso

E un tipo di dati che il sistema di gestione di Workflow non tratta mai. Sono dati strettamente relativi alle applicazioni che entrano in gioco nella realizzazione delle istanze di attività. Possono diventare pertinenti per il processo se sono utilizzate per determinare il cambiamento di stato di un'istanza d'attività.

Dati di controllo dei casi d'esecuzione (Workflow Process Control Data)

a) definizione

I dati di controllo dei casi d'esecuzione sono i dati gestiti dal sistema di gestione di Workflow e/o i motori di Workflow.

b) Uso

I dati di controllo dei casi d'esecuzione possono:

- . indicare lo stato corrente di un caso d'esecuzione
- . indicare lo stato di un'istanza d'attività (attiva o inattiva)
- . ecc...

c) Sinonimi

Dati di stato del WorkFlow

Dati di controllo del WorkFlow

Dati pertinenti per i processi (Workflow Process Relevant Data)

a) definizione

I dati pertinenti per i processi sono i dati utilizzati dal sistema di gestione di Workflow per determinare le condizioni di transizione ed il controllo degli itinerari dei casi d'esecuzione.

b) Uso

Dato che deve essere messo a disposizione di ogni attività ulteriore, e che può dunque eventualmente avere un'incidenza sulla scelta dell'attività ad effettuare.

Questi dati possono essere dati di decisione o valori di riferimento da trasmettere.

Storico (*Audit Trail*)

a) Definizione

Lo storico di un caso d'esecuzione è il rilevamento degli stati transitori dalla sua inizializzazione fino alla sua realizzazione finale.

b) Uso

esempio d'informazione che può essere raccolta in uno storico: data, ora e tipo di compito realizzato per ogni istanza d'attività.

3.3.9 Interfaccia e sviluppo

WAPI: Interfaccia di Programmazione d'Applicazione

Workflow (*Workflow Application Programming*

Interface/Interchange)

a) Definizione

l'interfaccia di programmazione d'applicazione Workflow funge da interfaccia per le applicazioni e strumenti Workflow verso il sistema motore Workflow. **WAPI** è un'iniziale per “Workflow Application Programming interfacciaInterchange”

b) Uso

Esempio : Un'applicazione deve passare per l'interfaccia di programmazione d'applicazione Workflow per richiedere buoni di lavoro al sistema di gestione di Workflow.

c) Sinonimi

- . API del Workflow
- . API del sistema di gestione di Workflow.

Interoperabilità del Workflow (Workflow Interoperability)

a) definizione

L'interoperabilità del Workflow è la possibilità che hanno molti motori Workflow di comunicare per effettuare in modo coordinato dei casi di processi su tutti questi vari motori.

b) Uso

- Possibilità d'interoperabilità tra motori eterogenei/omogenei.
- Possibilità di fare funzionare due motori come uno solo.

3.4 Alcuni termini inerenti alla piattaforma .NET usati in questa tesi

Termine	Definizione
. NET Framework	Il .NET Framework è una piattaforma per la realizzazione della nuova generazione di applicazioni e servizi Web XML distribuiti. Espone un modello di programmazione coerente, indipendente dal linguaggio e condiviso da tutti gli strati di un applicazione e consente di interoperare in modo trasparente e di migrare facilmente a partire dalle tecnologie esistenti. Il .NET Framework è costituito da tre componenti fondamentali: il Common Language Runtime, le Classi

	unificate ed ASP .NET.
ADO .NET	La tecnologia del .NET Framework per l'accesso ai dati.
API Web	API che consentono l'integrazione di un servizio Web XML all'interno del .NET Framework.
ASP .NET	La tecnologia Active Server Pages per il .NET Framework.
Assembly	L'unità per il deployment ed il controllo dei conflitti di versione del .NET Framework. Definisce gli spazi dei nomi per soddisfare le richieste e determina quali risorse esporre esternamente e quali rendere, invece, accessibili esclusivamente dall'interno dell'assembly. Un assembly include un manifesto che ne descrive i contenuti.
C#	Il primo linguaggio orientato alle componenti della famiglia C/C++. E' stato sottoposto all'ECMA per la standardizzazione.
CGI	Common Gateway Interface, il primo protocollo Internet utilizzato per generare contenuti interattivi sul Web.
CLR	Common Language Runtime. I sistemi dei tipi dei metadati e di esecuzione forniti dal .NET Framework, che forniscono codice gestito e dati con servizi quali l'integrazione cross-language, la sicurezza per l'accesso al codice, la gestione del tempo di vita degli oggetti ed il supporto per il debugging ed il profilino.
DOM	<i>Document Object Model</i> , è una rappresentazione in memoria di un documento XML; grazie ad essa è possibile leggere e modificare un documento XML a livello di programmazione.
DTD	<i>Document Type Definition</i> , un file che indica le regole per la definizione e la correlazione logica tra elementi, gli attributi e gli altri dati di un documento XML.
Form Web	Le <i>form Web</i> sono una tecnologia ASP .NET utilizzabile per realizzare pagine Web programmabili. Le Form Web possono visualizzare informazioni all'utente, utilizzando un qualsiasi linguaggio di marcatura, in qualsiasi browser ed utilizzare il codice server per implementare la logica applicativa.
GPL	<i>General Public Licenses</i> , licenza che garantisce l'acquisizione e la libera circolazione di Software libero utilizzabile e modificabile di chiunque senza scopo di lucro.
HTML	<i>Hyper Text Markup Language</i> , nasce come lingua franca per pubblicare informazioni ipertestuali sul World Wide Web. L'HTML è un formato, non proprietario, basato su

	SGML (Standard Generalized Markup Language).
HTTP	<i>Hyper Text Transfer Protocol</i> , protocollo Internet standard per trasferire informazioni tra Client e Server, Server e Server.
IDL	Interface Definition Language, un linguaggio utilizzato dall'applicazione per specificare le interfacce che intende esporre ad altre applicazioni.
IL	<i>Intermediate Language</i> , è un linguaggio utilizzato come input per numerosi compilatori e come input per un compilatori JIT. L'IL definisce un'architettura di esecuzione astratta e basata sullo stack.
JIT	<i>Just In Time</i> , una fase che descrive un'azione che viene eseguita solo quando necessario, come la compilazione Just in Time o l'attivazione dell'oggetto Just IN Time. Per convenzione il termine JIT viene utilizzato per riferirsi al compilatore JIT.
Metadati	Informazioni sui dati. I metadati vengono memorizzati assieme a questi all'interno dei file eseguibili ed utilizzati da compilatori, strumenti e dal runtime per fornire un ricco insieme di servizi.
Proxy Class	Classe C# autogenerata dal tool wsdl.exe di .NET Framework che consente l'accesso ai dati forniti da un XML Web Service.
SOAP	<i>Simple Object Access Protocol, standard W3C</i> . Un protocollo leggero per lo scambio di informazioni in un ambiente decentralizzato e distribuito. E' un protocollo basato su XML costituito da tre parti: uno strato esterno che definisce una struttura per la descrizione e l'elaborazione del contenuto di un messaggio, un insieme di regole di codifica per esprimere istanze di tipo dato ed una convenzione per la rappresentazione di invocazioni a procedure remote relativamente esposte.
UDDI	Le specifiche <i>Universal Description Discovery and Integration</i> , un'iniziativa che crea una struttura aperta, globale ed indipendente dalla piattaforma che consente ai servizi aziendali di individuarsi gli uni rispetto gli altri, definire le modalità di interazione su Internet e condividere informazioni in un registry globale.
WebMethod	Parola chiave del .NET Framework che consente di accedere agli oggetti da Internet.
WSDL	<i>Web Service Description Language</i> , è una grammatica XML utilizzabile da sviluppatori e strumenti di sviluppo per rappresentare le funzionalità di un XML Web Service.
XML	<i>Extendible Markup Language</i> , è uno standard W3C per la formattazione di documenti e dati strutturati sul Web.
XML Web Service	Un servizio Web XML è un applicazione che espone le proprie funzionalità da codice su Internet utilizzando

	protocolli standard come XML, HTTP, SOAP e MIME.
XPath	<i>XML Path Language</i> , è il risultato di uno sforzo per fornire di una sintassi comune tra XSLT e XPointer; lo scopo primario di XPath è quello di indirizzare parti di un documento XML e fornire funzioni di base per la manipolazione di numeri, stringhe e boolean.
XSD	Schema Standard W3C che codifica un unico documento di riferimento che funge da prototipo. In questo modo possono essere definite una volta per tutte sia la struttura che la grammatica degli elementi che compongono tutti i documenti XML scambiati.
XSL	<i>Extensible Stylesheet Language</i> , serie di tag che possono essere usati per applicare delle regole di formattazione a ciascuno degli elementi interni ad un documento XML.

4. K2.NET 2003: PRESENTAZIONE E PROSPETTIVA TECNICA

4.1 Presentazione del Prodotto

K2.net 2003 è il primo strumento di interazione **uomo-sistema** e **sistema-sistema** al livello dell'impresa, permettendo l'automatizzazione veloce dei processi mediante l'integrazione di soluzioni di livello applicazione ed utente. K2.net 2003 è un insieme di soluzioni personalizzabile che si adatta in modo dinamico al modo in cui le soluzioni di applicazioni di impresa sono sviluppate.

EBPM(Enterprise Business Process Management) gestisce in modo sistematico tutti i processi di impresa per integrarli alla Definizione dell'obiettivo strategico.

K2.net 2003 propone una piattaforma di collocamento che opera a livello di EBPM, permettendo all'impresa di continuare a rispondere ai bisogni dei suoi clienti senza preoccuparsi degli eventuali guasti del sistema o interruzioni del servizio. Grazie a K2.net 2003, l'organizzazione può gestire efficacemente i suoi processi e può assicurare un servizio continuo.

K2.net 2003 permette ai processi di evolversi nel loro ambienti naturali, senza ricorrere ai cicli di collocamento in corrispondenza di processo di impresa a lungo termine. La gestione di versioni per le soluzioni permette all'utente di conservare una traccia delle differenti soluzioni, ciò che si rivela utile per risolvere i problemi dell'impresa.

La soluzione K2.net 2003 comunica tra componenti con l'aiuto di servizi Web, come il SOA(Servizio Oriented Architecture). Inoltre, SourceCode ha preso la misura dello scarto rispetto alle architetture basate sugli eventi, (Event Driven Architetture, EDA) secondo che una combinazione di servizi e di motori di eventi pilota i processi basati sugli eventi

4.2 K2.net 2003 : Caratteristiche generali

- **K2.net uses an event-driven workflow engine that provides an environment where programs and processes can run seamlessly.**
- **K2.net is an enterprise human-to-human and human-to-system workflow platform built on .NET for .NET.**
- **K2.net automates business processes spanning people, technology and distance using the .NET platform.**
- **K2.net provides the underlying infrastructure, built on the .NET platform, to integrate, leverage and optimize the investment in technology infrastructure.**

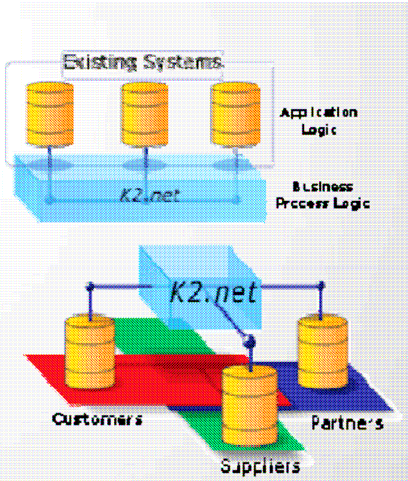


Figura 4.1: K2.net 2003 : Caratteristiche generali

- ⇒ K2.net usa un motore di flusso di lavoro “event-driven” che fornisce un ambiente dove i programmi ed i processi possono essere eseguiti perfettamente.
- ⇒ K2.net è un piattaforma Workflow (uomo-a-uomo e uomo-macchina) costruito su .NET per il .NET
- ⇒ K2.net automatizza i processi di affari attraverso le persone, la tecnologia e la distanza usando la piattaforma .NET
- ⇒ K2.net fornisce l'infrastruttura sottostante, costruita sulla piattaforma .NET, per integrare, la leva finanziaria ed ottimizzare l'investimento nell'infrastruttura di tecnologica.

4.3 Ciclo di vita dei processi d'impresa

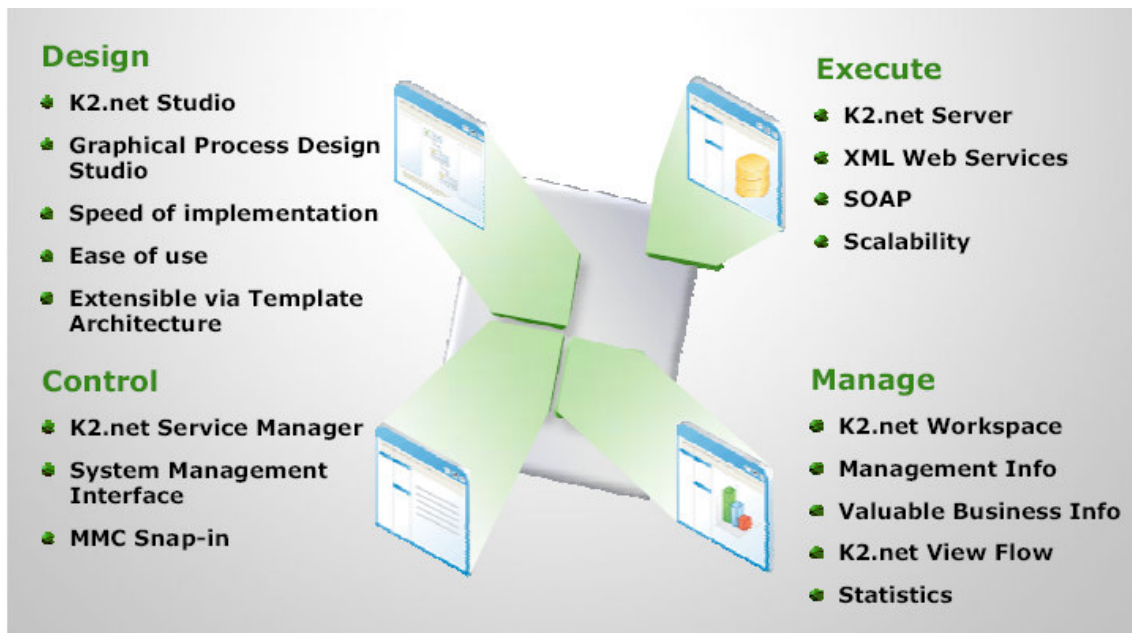


Figura 4.2: ciclo di vita di un processo d'impresa

4.3.1 Design (*Disegno*)

Il processo impresa per essere automatizzato è disegnato con un programma grafico per processi, chiamato **K2.net 2003 Studio**. L'interfaccia grafica (tramite lo strumento drag-and-drop simile al copia incolla per trascinamento in ambiente windows) consente un rapido sviluppo di processi workflow, tenendo conto dell'automazione rapida del processo.

4.3.2 Execute (*Esecuzione*)

Il processo automatico d'impresa è eseguito da **K2.net 2003 server**. Il server applica le varie regole d'impresa che sono state costruite nel processo durante la sua definizione (disegno con **K2.net 2003 Studio**). Il server espone anche i servizi di XML, consentendo la facilità di integrazione con altri programmi (software) d'impresa.

4.3.3 Control (controllo)

Il controllo in questo caso fa riferimento all'amministrazione di server insieme ai processi gestiti. Questa funzione amministrativa è eseguita usando il **K2.net 2003 Manager** di

Servizio. Il manager è la console di Gestione di Microsoft (MMC - Microsoft Management Console snap-in)

4.3.4 Manage(Gestione)

Una parte vitale del ciclo vita del processo è il controllo(monitoraggio) dei processi. I processi sono controllati per rivelare qualunque collo di bottiglia o altri ritardi nel ciclo di esecuzione del processo. Il **K2.net 2003 Workspace** (Area di lavoro del K2.net 2003) è l'interfaccia utente che dà ad un operatore la capacità di avere le informazioni dettagliate sui processi in esecuzione o completati. Offre all'utente una gestione preziosa delle informazioni d'impresa.

4.4 Architettura e principali componenti di K2.net 2003

4.4.1 I vari livelli

L'architettura generale del software di K2.net è illustrato nella figura di sotto. I vari componenti saranno descritti nei paragrafi successivi. Come si vede l'architettura è composta da quattro livelli. Il **primo livello** chiamato *Client Layer(livello utente)* è l'interfaccia tra le applicazioni del workflow e gli utenti(Programmatori o attori del workflow). Il **secondo livello** rappresenta i **Modelli degli oggetti** (Object Model) che consentono alle applicazioni di K2 oppure applicazioni esterne di interagire con il server. Il terzo livello chiamato **Livello Server**(Server Layer) è l'architettura hardware/software del (o dei) server dove sono installati i componenti server di K2.net. L'ultimo livello chiamato **Livello Dati**(Data Layer) è il tipo di database usato per memorizzare i dati del workflow.

La possibilità di distribuire i lavori (Work item) attraverso la LAN, WAN e INTERNET assicura che tutti i partecipanti(attori) di progetto di Workflow,di ricevere il loro lavori in tempo. Tutti i dati , fasi, struttura del progetto workflow sono memorizzati in due database gestiti da K2.net

L'uso di un sistema di gestione di base di dati relazionale di contenere l'insieme dei lavori assicura che i dati immagazzinati possono crescere con gli aumenti di richiesta e volume.

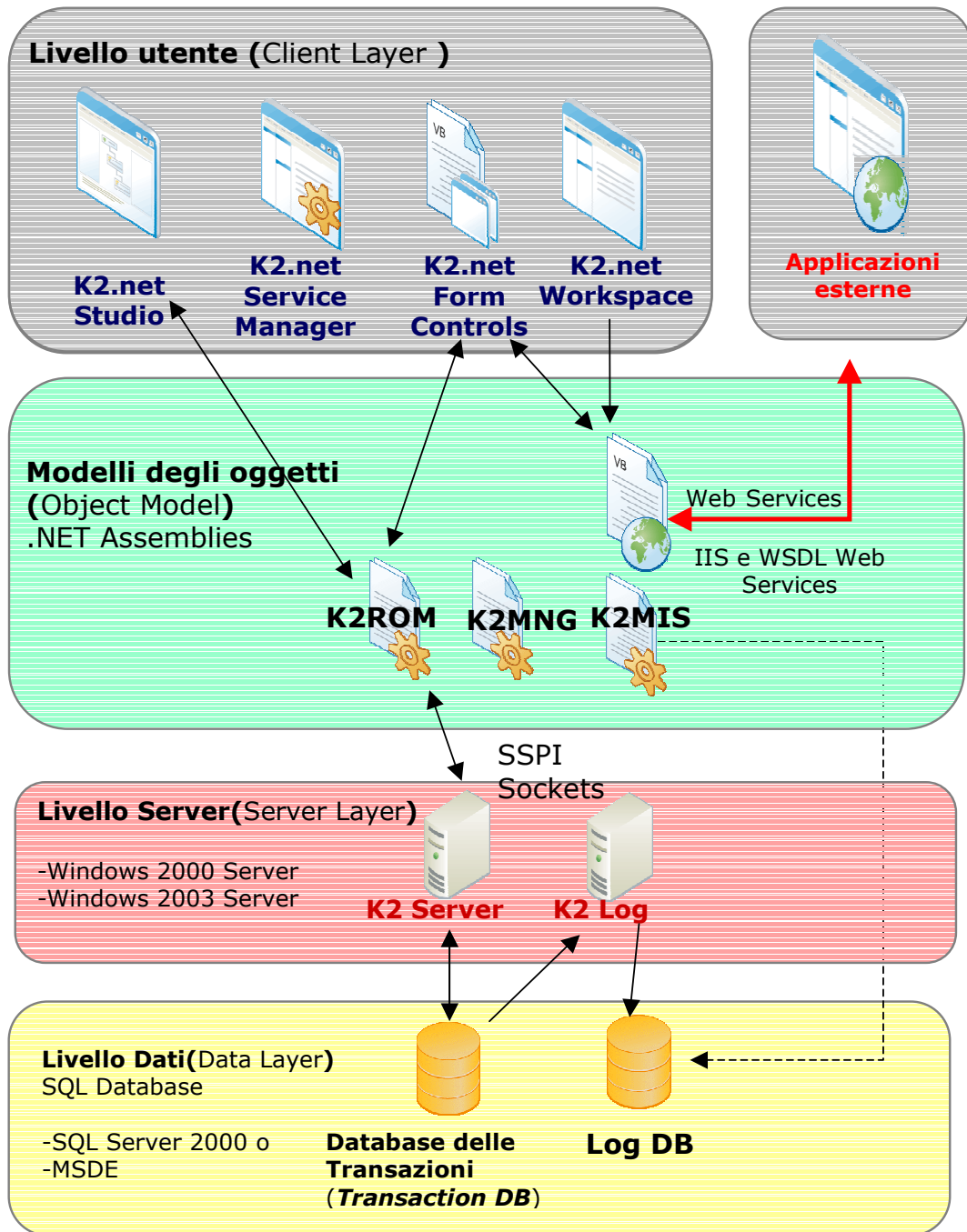


Figura 4.3: K2.net 2003: Architettura

4.4.2 Architettura k2.net : K2.net 2003 Studio

K2.net 2003 Studio è un potente ambiente di progetto di flusso di lavoro. Collega le persone, applicazioni, e informazioni nei processi d'impresa automatici. Serve alla concezione del processo e semplifica considerevolmente il suo collocamento in opera e la sua automatizzazione.

Principali caratteristiche:

- ❑ Ambiente di concezione di flusso di lavori di facile utilizzo per l'utente, facilitando la concezione e la creazione delle soluzioni di flusso di lavori completi(compresi Processi, utenti, gruppi, attività, eventi, dati, orario di lavoro...).
- ❑ K2.net Studio comunica con K.net server solo via gli object model (chiamato anche K2.net 2003 API). Usa la porta predefinita 5252 durante la comunicazione
- ❑ I processi concepiti in K2.net 2003 studio possono essere estesi e personalizzati per mezzo del codice VB.NET e C#.
- ❑ Integrazione assistita del processo ai seguenti elementi :
 - Formulari HTML ASP.NET
 - Documenti dinamici Microsoft Office (Formulari InfoPath 2003)
 - Microsoft SharePoint
 - Microsoft Exchange et Microsoft Outlook
 - Posta SMTP
 - Microsoft MSN Messenger
 - Piattaforme di diffusione mobile (Pocket PC e Tablet PC)

4.4.3 Architettura k2.net : K2.net 2003 SmartForms

Il K2.net 2003 SmartForm controls per ASP.NET è una serie di interfaccia di form e interfaccia utente che consentono agli operatori di costruire rapidamente il flusso di lavoro(attraverso I form), che possono funzionare in ambiente ASP.NET e accessibili via browser su internet

Principali caratteristiche:

- ❑ Creazione di formulari Internet ASPX che permettono di collegare le persone e le applicazioni d'impresa ai processi.
- ❑ I controlli estendono la portata di Microsoft ASP.NET Framework.
- ❑ Fornitura di un insieme di controlli, di “assembly” .NET e di modelli.

- ❑ Possibilità d'elaborazione in un qualunque editore compatibile con ASP.NET, tra cui Microsoft Visual Studio.NET.
- ❑ Linguaggi compatibili:
 - Visual Basic .NET
 - C#

4.4.4 Architettura k2.net : K2.net 2003 Service Manager

Facilita l'amministrazione generale del sistema e consente alle soluzioni automatizzate di flusso di lavoro di funzionare senza collisioni. Da qui è possibile assegnare i vari permessi ai vari attori del workflow.

Integrazione a InfoPath:

- ❑ Conversione di formulari InfoPath esistenti ai flussi di lavori per mezzo di modelli proposti da un assistente.
- ❑ Controllo contestuale della vista del flusso di lavoro, permettendo di presentare all'utente la vista adeguata del documento InfoPath ad ogni fase del flusso di lavoro.
- ❑ Migrazione della gestione delle conoscenze, dalla carta verso un supporto elettronico.

4.4.5 Spazio di lavoro K2.net 2003 (K2.net 2003 Workspace)

K2.net 2003 Workspace consente agli utenti di visualizzare (la lista dei lavori "Worklist"), controllare l'andamento dei dati associati ai processi attraverso rapporti. Questo permette di analizzare, migliorare i processi evitando anche i colli di bottiglia.

Principali caratteristiche:

- ❑ Interfaccia web personalizzabile, che permette agli utenti di separare e filtrare in modo dinamico l'elenco di lavoro e di effettuare relazioni.
- ❑ Informazioni dettagliate sull'attività
- ❑ Grafico in tempo reale delle istanze di processi in corso tramite K2.net2003 View Flow.
- ❑ Recupero e nuova assegnazione dinamici dei compiti.
- ❑ Possibilità di definire condizioni d'assenza ed di delegazione dei compiti di conseguenza.

4.4.6 Architettura k2.net : Integrazione a BizTalk Server

Principali caratteristiche:

- ❑ Forte integrazione all'infrastruttura d'adattamento BizTalk tramite adattatori specifici a K2.net assumendosi le esigenze dei flussi di lavori tra sistemi e tra imprese.
- ❑ Appoggio sull'infrastruttura di adattatori esistenti in processo d'impresa per un'integrazione sistema a la scala dell'impresa.
- ❑ Integrazione al motore delle norme d'impresa, che permette di definire norme di flusso di lavori nell'ambito di una banca centrale di regole.

4.4.7 Architettura k2.net: K2.net 2003 Runtime Object Model

K2.net 2003 Runtime Object Model propone un insieme completo di API (che consente a **K2. Net Studio di comunicare con K2.net Server oltre che ad applicazioni esterne**) con “assembly .NET” e servizi web WSDL.

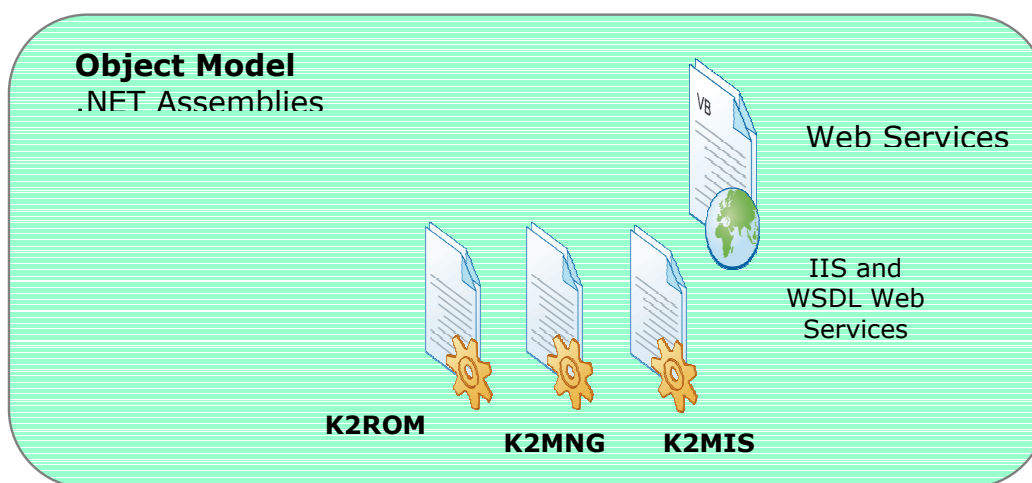


Figura 4.4: Runtime Object Model di K2.net 2003

4.4.7.1 K2MNG

K2 Management object espone le funzioni del Service Manager per gli accessi programmati alle funzioni amministrative. Per esempio è possibile accedere alla lista dei Processi esportati sul K2.net 2003 server con i Metodi GetProcessFolders() e GetProcSets() usando l'utente amministratore di K2.net 2003 server per il login.

4.4.7.2 K2MIS

Il Workspace object espone le funzioni del Workspace per gli accessi programmati alle funzioni di report.

4.4.7.3 K2ROM (Runtime Object Model)

Le Object Model rappresentano k2.net object come una collezione di classi. K2.net 2003 ROM è un assembly .NET che espone k2.net Server al mondo esterno usando oltre agli assembly o WSDL web services. la ROM comunica con K2.net Server usando sockets con SSPI o SOAP (Web Services)

Di seguito c'è un breve insieme di funzionalità esposta dal modello di oggetto:

- avviare un processo workflow in modo programmato
- Accesso programmato alla lista di lavori (work list items).
- Accesso programmato alle istanze dei processi
- Accesso programmato alle istanze delle attività
- Accesso programmato alle istanze degli eventi

4.4.8 Architettura k2.net: K2.net 2003 Server

In K2.net 2003 Server sono memorizzati la struttura e definizione dei processi e motore delle soluzioni dei flussi di lavoro.

Principali caratteristiche:

- Gioco di strumenti integrabili che favoriscono la gestione di server centralizzata.
- K2.net il 2003 Server è senza stato, cioè tutti i dati riguardanti lo stato si trovano in SQL Server.

- Presa a carico dell'autenticazione Kerberos (Sistema di sicurezza che usa la crittografia per impedire di scoprire password e penetrare file e cartelle) su molti settori.
- Integrazione e presa a carico di Microsoft Active Directory Services (ADS): utilizzo delle strutture gerarchiche definite in ADS, senza impatto su struttura ADS né Definiziones di schemi, con collegamento con il deposito ADS.
- Destinazioni dinamiche e segnaletica progressiva effettuate da K2.net 2003 Server per assicurarsi che il compito è trasmesso tempestivamente al partecipante interessato e che è effettuata prima della scadenza.
- Architettura di thread libera ed ottimizzazione dell'equilibrio di carico:
- K2.net il 2003 Server utilizza Microsoft SQL Server 2000/2005 come banca di dati sottostante, ciò che permette di beneficiare di una piattaforma molto evolutiva ed affidabile.
- Facilità di gestione grazie ad un ambiente di gestione centralizzata per i server K2.net 2003 distribuiti, per mezzo della rete locale, estesa o privata virtuale dell'impresa o per Internet.

4.5 Compatibilità con le norme professionali (BML, Wf-XML e BPEL)

K2.net 2003 rispetta le norme seguenti:

- WFMC (Workflow Management Coalition): SourceCode è membro fondatore del WFMC e cerca di contribuire alle norme WFMC e rispettarle.
- SOAP: K2.net 2003 prende carico la versione Microsoft.NET di SOAP tramite biblioteche.NET e del motore d'esecuzione. K2.net 2003 permette di utilizzare in gran parte i servizi web basati su SOAP tramite K2.net2003 studio e K2.net2003 Server. Così, i progettisti di processi possono aggiungere riferimenti ai servizi web SOAP nelle loro soluzioni di flusso di lavori.
- XML/XSD: K2.net 2003 utilizza in gran parte gli schemi XML e XSD per Definizione dei processi, la gestione dei dati e la comunicazione.

- OLE/COM (Object Linking and Embedding/Component Object Model): K2.net 2003 studio possiede un API d'automazione COM integrata, permette così di sviluppare ed estendere le soluzioni per mezzo del modello oggetto d'automazione(sarà usato nel corso di questo studio per fare interagire il software esterno con k2.net 2003). Questo consente di utilizzare i modelli d'oggetto d'automazione che esistono in Microsoft Word e Microsoft Excel. Il modello d'oggetto dell' API K2.net 2003 è una versione d'assembly .NET e non è disponibile come interfaccia COM.
- ODBC (open Database Connectivity): K2.net 2003 è compatibile con tutti i fornitori di dati.NET, tra cui ODBC, ADO.NET ed il fornitore Oracle.NET.
- VIM (Vendor Independent Messaging): la messaggeria VIM è presa in carica attraverso la versione COM e la istanziazione degli oggetti del fornitore.
- MAPI (Messaging Application Programming Interface): K2.net 2003 è interamente compatibile con MAPI.
- X.400 e X.500 : norme internazionali di comunicazione per la posta elettronica ed i servizi d'elenco: K2.net 2003 è interamente compatibile con X.400 e X.500

4.6 Gestione dei processi distribuiti

La gestione dei processi distribuita è una funzione nativa di K2.net 2003. Permette ad un utente di esportare lo stesso processo verso ambienti multipli ed effettuarlo simultaneamente su server distinti, se necessario.

La funzione di raggruppamento permette di migliorare la capacità di trattamento negli ambienti d'impresa e garantisce una tutela tramite cambiamento per ridurre al minimo i rischi nel caso di uno spegamento a grande scala. È possibile configurare come server K2.net2003 Server fino a 32 elaboratori(server). Questa funzione di raggruppamento , che appartiene a K2.net 2003, non somiglia alla tecnologia di cluster¹ di Microsoft. Un'altra differenza risiede

¹ Uno o più terminali collegati a una unità di controllo delle comunicazioni su una linea di trasmissione

nel fatto che ogni nodo di un albero K2.net 2003 può possedere diverse specificazioni fisiche ed essere gestito ed essere configurato separatamente in funzione della capacità del materiale.

K2.net 2003 propone una funzione IPC (Inter processo comunicazione) che permette ai processi principali di chiamare sotto-processi. Questa funzione permette a K2.net2003 di chiamare, controllare e gestire processi su molte macchine. Questi processi diventano sensibili ai dati ed agli stati a il loro integrazione. In altri termini, obbediscono sotto forma di processo. I processi distanti possono essere chiamati in modo sincrone o asincrono.

4.7 Architettura dell' applicazione

4.7.1 Porta e protocollo di rete

K2.net 2003 utilizza la porta 5252 . Questo numero, può essere cambiato nel caso di necessità. K2.net utilizza sia le classi .NET framework, sia le DLL del client SQL nativo. Il protocollo di comunicazione utilizzata tra le DLL ed il server SQL fisico costituisce una parte dell'impianto del client SQL sull'elaboratore in questione. L'amministratore di base di dati SQL può allora configurare gli strumenti del client per potere utilizzare una porta qualunque senza impatto su K2.net.

4.7.2 Autenticazione degli utenti

Nello standard, K2.net 2003 prende in carico l'autenticazione NTLM²(Windows NT LAN Manager) e Kerberos sui domini di Microsoft active Directory. I domini multipli sono anche presi in carico se la comunicazione tra rete adeguata è attivata.

4.7.3 Presa in carico del XML per la definizione dei processi

K2.net 2003 è interamente compatibile con la XML. I campi di dati XML possono costituire in parte la definizione del processo. Possono essere creati manualmente o gli schemi possono essere importati da una fonte esterna (Esempio: un documento InfoPath). Internamente, le definizioni dei processi K2.net 2003 sono conservate in XML per mezzo di uno schema XML definito. Le definizioni di dati dei processi in corso sono conservate e messe a disposizione in XML.

² NTLM è il protocollo predefinito di autenticazione di rete nel sistema operativo Windows NT 4.0

4.7.4 Integrazione dei dati

Nell'ambito di un processo K2.net2003, esiste anche un controllo a livello dell'attività ottenuto creando un numero di serie unica per istanza di un evento. In altri termini, K2.net Server garantisce il controllo e la gestione di un processo attraverso i suoi cambiamenti di stato, cosa che permette di assicurarsi che un processo sia seguito nel corso del suo ciclo di vita. Così, non è possibile a due utenti lavorare su uno stesso elemento a partire da due periferiche distinte ed a momenti diversi.

Gli eventi IPC (Inter-process communication) possono anche aggiornare/copiare valori di dati tra processi. I campi di dati di un processo sono definiti al momento della concezione e possono conservare la maggior parte dei tipi di dati correnti, comprese le informazioni binarie e XML.

4.7.5 Gestione e risoluzione degli errori

Lo strumento di riparazione degli errori in tempo reale permette agli amministratori di effettuare modifiche per risolvere errori e riprendere l'esecuzione con minori danni possibili. Un amministratore può configurare il server affinché ripeta una sezione un certo numero di volte prima di segnalare un errore. In questa fase, il codice può essere eseguito di nuovo (Esempio: se l'errore è dovuto ad un fattore esterno che è stato risolto) o può essere modificato prima di essere eseguito di nuovo (se l'errore è stato causato da un problema del codice stesso, da Esempio).

Inoltre, se lo stesso problema si presenta più volte, l'amministratore ha la possibilità di riparare tutte le istanze in un solo passaggio, cosa che permette di accelerare considerevolmente la ripresa in caso di guasto importante.

4.7.6 Gestione delle versioni

4.7.6.1 Versione e gestione delle modifiche

La gestione delle modifiche durante la fase di concezione del processo è delegata a strumenti di terzi.

K2.net 2003 prende in carico tutti i sistemi di controllo delle modifiche atte a gestire i tipi di archivi binari, come Microsoft Visual SourceSafe.

4.7.6.2 Versioni multipli dello stesso processo di un flusso di lavoro eseguito simultaneamente

K2.net Server può gestire molte versioni di un processo. Inoltre, molte versioni di un processo possono essere attivate simultaneamente. Per principio, un'istanza di processo esistente si conclude con versione con la quale è cominciata. I nuovi processi cominciano con la nuova versione attiva del processo. È anche possibile ritornare a versioni precedenti dei processi, sia su una base ad hoc, in occasione della creazione di un'istanza di processo, cioè globalmente, tramite l'interfaccia K2.net Service Manager.

I server K2.net 2003 sono registrati tramite comandi RPC, con collegamenti di socket SSPI, cosa che permette di accedere ai server K2.net 2003 in qualsiasi momento e ovunque

4.8 Posizione di K2.net 2003 in uno spazio BPM (Business Plan Management)

Lo spazio del BPM (Business Process Management), come presentato nella figura successiva, può essere suddiviso in tre livelli di Workflow chiamati, Uomo-a-Uomo(*human-to-human*), Uomo a sistema(*human-to-system*) e sistema - sistema (*system-system workflow*).

4.8.1 Percorso Uomo-a-Uomo (*Human-to-human routing*)

Questo livello di flusso di lavoro consiste in un percorso di un documento da un operatore a un altro utilizzando un sistema di e-mail. Questo tipo di flusso di lavoro non è gestito di solito da regole molto strette d'impresa. La capacità di verificare (controllare) questo tipo di flusso di lavoro è molto limitato o inesistente.

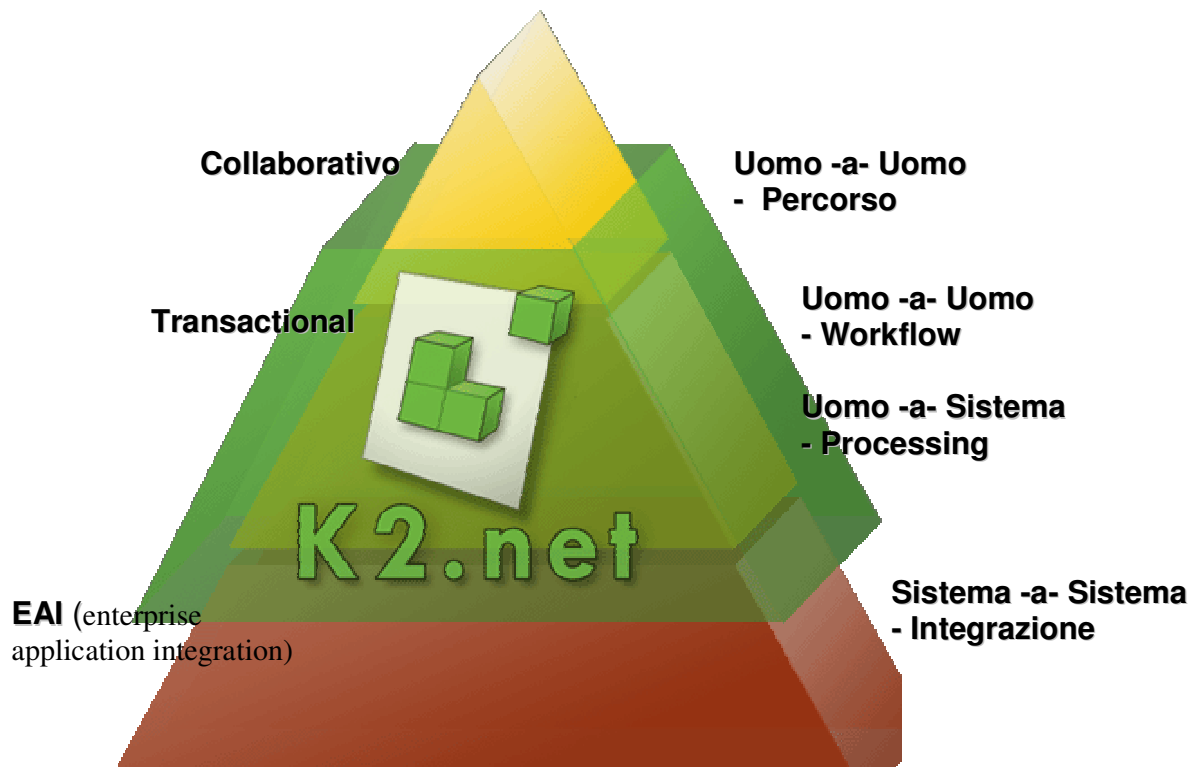


Figura 4.5: K2.net 2003 nello spazio del Business Process Management

4.8.2 Workflow Uomo-a-Uomo e Uomo-a-sistema (*Human-to-human and human-to-system workflow*)

Questo livello di workflow è solitamente gestito da regole rigide dell'impresa (*Business rules*). Le regole dell'impresa (*Business rules*) sono eseguite da un tipo di motore di workflow. Ogni azione realizzata (eseguita) da un utente su un processo è vista una transazione e salvata in qualche modo. Questo tipo di workflow si presta ai processi automatizzati. È inoltre molto probabile, in questo scenario, che un insieme di dati siano passati ad applicazioni o servizi per essere processati prima di essere passati al prossimo utente. Questo si riferisce ad un Workflow Uomo-a-sistema. . K2.net 2003 è una soluzione che opera a questo livello.

4.8.3 Workflow sistema-a- sistema (*System-to-system workflow*)

Si parla di workflow sistema-a-sistema quando abbiamo un certo numero di applicazioni commerciali/gestionali che devono scambiarsi i dati per raggiungere un determinato obiettivo. Questo tipo di Workflow è spesso implementato durante il progetto d'integrazione delle applicazioni d'impresa (EAI). Microsoft's BizTalk server opera a questo livello.

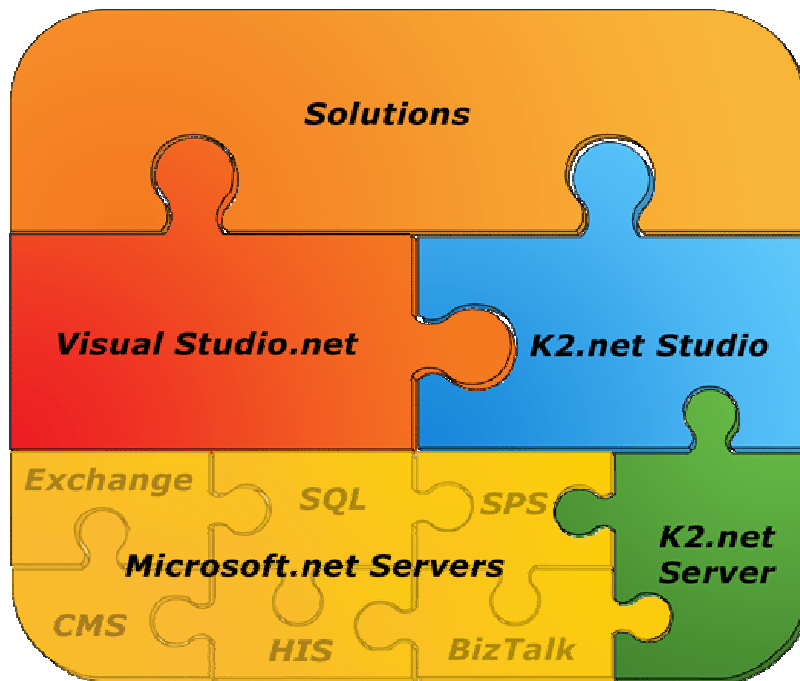


Figura 4.6: Posizione K2.net 2003 nello spazio delle soluzioni Workflow

In questa mappa sintetica vediamo come i vari componenti di K2.net 2003 si integrano con i vari componenti software dell'ambiente Microsoft questo per ottimizzare i tempi di sviluppo delle soluzioni workflow.

Come si vede K2.net 2003 Studio è modellato intorno a Visual Studio.NET per consentire agli sviluppatori di creare in modo efficiente e rapido le soluzioni Workflow basati su .NET

5. K2.NET 2003 STUDIO : STRUTTURA DI UNA SOLUZIONE WORKFLOW

5.1 Introduzione

Il componente K2.net Studio viene usato per disegnare tutta struttura della *Soluzione* di in sistema di Workflow. Dopo la fase di progettazione di un sistema Workflow , bisogna disegnare tutta la struttura della Soluzione, Progetto e Processi con i vari diagrammi di flussi usando K2.net Studio. Da questo componente vengono definiti le variabili, le proprietà(ora di esecuzione, durata,...), gli utenti dei vari Processi e Attività. Dopo la sua implementazione la Soluzione viene esportato sul server K2.net

5.2 Struttura generale di una Soluzione

Di seguito c'è lo schema generale di una soluzione che può essere disegnato da K2.net Studio

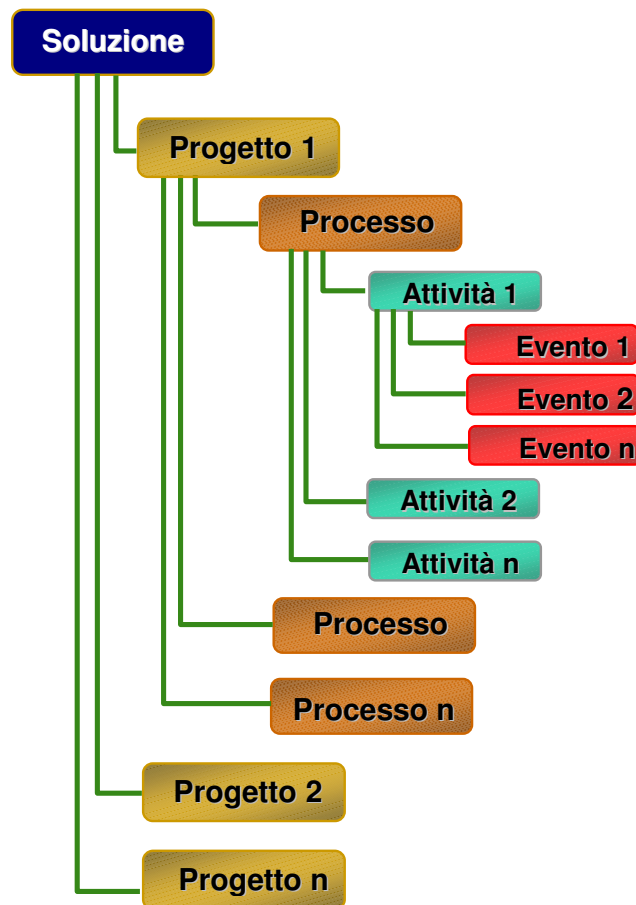


Figura 5.1: Struttura generale di una Soluzione Workflow vista da K2.net 2003

Dalla Grafica si vede che una Soluzione (Solution) è composta da uno o più Progetti(Project). Ogni Progetto a sua volta è composto da uno o più processi (Process) con varie attività (Activity). Ogni attività di un determinato processo può generare un o più eventi (events) , ad esempio l'invio di una mail o richiesta dati ad un determinato utente.

la Solution(Soluzione creata da K2.net 2003 Studio) è in un insieme di file memorizzati all'interno di una cartella con il nome della Solution. All'interno della cartella si trova il file con il nome della solution e l'estensione .ksn.

I progetti e Processi sono anche creati all'interno della cartella della Solution e tutte le informazioni del progetto e Processi sono salvati in due file con estensione *.k~~p~~j per i progetti e .kpr per i processi .

5.3 Struttura generale di un Processo Workflow disegnato con K2.net 2003 Studio

Esempio:

In questo esempio un utente (impiegato) fa una richiesta di acquisto di un materiale (informatico o non) via internet/intranet. Se il materiale è informatico in questo caso la richiesta viene presa in carico dal servizio informatico. Tale servizio controlla se il materiale (esempio Tastiera, monitor...) è già presente in azienda. Se non è presente inoltra la richiesta al responsabile che deciderà se il materiale sarà acquistato o no.

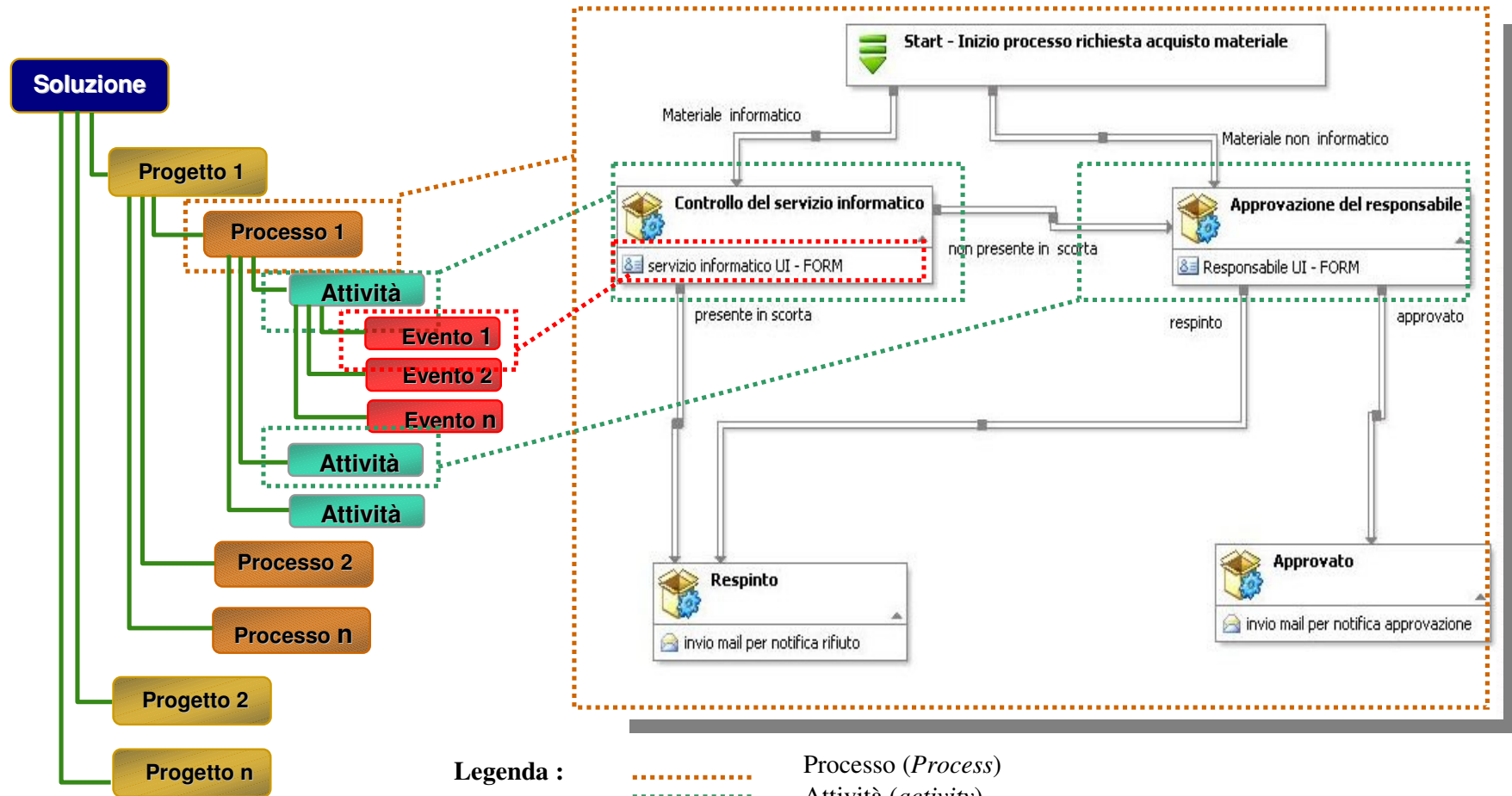


Figura 5.2: Struttura di un processo visto da K2.net 2003 Studio

5.3.1 Creazione di una soluzione con K2.net 2003 Studio

Durante la creazione di una nuova soluzione(*solution*), progetto o processo in K2.net Studio, saranno richiesti le proprietà di base (esempio: nome e percorso dove salvare la soluzione).

- Per le soluzioni è necessario inserire solo il nome e il percorso durante la creazione.
- Per i processi, si può dopo la creazione, fare il tasto destro sul processo e selezione “properties” per modificare la proprietà.

Di seguito ci sono le varie proprietà delle Soluzioni, Progetti e Processi che possono essere impostati durante o dopo la creazione

Note: le proprietà indicate con * possono essere impostati solo durante la creazione e diventano di sola lettura

5.3.1.1 Proprietà della Soluzione (*Solution*)

- **Name***: È il nome della Soluzione. K2.net Studio crea automaticamente una cartella e un file con lo stesso nome della Soluzione.
- **Location***: È il percorso sul disco dove verrà creata la cartella della Soluzione. Il file della soluzione ha come estensione **.ksn**

5.3.1.2 Proprietà del Progetto (*Project*)

- **Name**: È il nome del Progetto. K2.net Studio crea automaticamente una cartella e un file con lo stesso nome del Progetto. Se il nome del progetto viene modificato in seguito il nome del progetto sarà visualizzato in K2.net Studio ma il nome originale della cartella del progetto sarà mantenuto.
- **Location***: È il percorso sul disco dove verrà creata la cartella del Progetto. Il file del Progetto ha come estensione **.kpj**.
- **Language***: linguaggio di sviluppo per la codifica che può essere **Vb.net** o **C#**.
- **Description**: Testo di descrizione del progetto.
- **Working Hours**: Permette di impostare l'orario di esecuzione dei processi. Per esempio evitare l'esecuzione durante il fine settimana.

- ❑ **Export Servers:** Specificare il server “K2.net 2003 server” dove si vuole esportare il progetto.
- ❑ **References:** Usato per riferimento ad altri .NET assemblies, COM objects e Web references.
- ❑ **Imports:** Usato per importare il *Namespace* nel progetto.

5.3.1.3 Proprietà Processi (Process)

- ❑ **Name:** E il nome del Processo. K2.net Studio crea automaticamente un file con lo stesso nome del Processo. Se il nome del progetto viene modificato in seguito il nome del progetto sarà visualizzato in K2.net Studio ma il nome originale della cartella del progetto sarà mantenuto.
- ❑ **Location*:** E il percorso sul disco dove verrà creata la cartella e il file del Processo. Il file del Processo ha come estensione **.kpr**.
- ❑ **Description:** Testo di descrizione del processo.
- ❑ **Priority:** High(alto), Medium(medio) or Low (basso). Impostando la proprietà si consente così all’utente di filtrare la lista di lavoro in base alla priorità.
- ❑ **Expected Duration:** Consente di impostare la durata del processo. Questo è molto importante per evidenziare tutti i processi non terminati entro il tempo prefissato.
- ❑ **Working Hours:** Si può scegliere di ereditare l’orario di esecuzione del Progetto a cui appartiene il processo oppure impostare il suo orario di esecuzione.
- ❑ **Data Fields:** Usato per definire le variabili del processo.
- ❑ **Exceptions:** Usato per specificare come saranno registrati gli errori in K2.net 2003 Server se il processo si interrompe.

5.3.1.4 Proprietà dell’Attività (Activity)

Le attività rappresentano un passaggio logico nel processo di flusso di lavoro. Le Attività in un Processo sono i punti in quali sono prese le decisioni, i dati sono consegnati e le azioni sono eseguite(eseempio invio di una mail) .Ecco di seguito alcune proprietà di base.

- ❑ **Name:** E il nome dell’Attività. Esempio e.g. Approvazione del responsabile.

- ❑ **Description:** Testo di descrizione dell'attività.
- ❑ **Priority:** Ogni attività di un processo può avere la sua priorità anche diversa da quello del processo.
- ❑ **Expected Duration:** Consente di impostare la durata presunta dell'Attività
- ❑ **Working Hours:** Ogni attività può avere il suo orario di esecuzione.
- ❑ **Destination Rule:** Imposta gli utenti coinvolti in questa attività. Se un'Attività contiene un evento utente (*client event*) allora è necessario impostare questo campo altrimenti la compilazione del processo provocherà un errore

Destination Rule Options:

- ❑ **Send To:** Spedisce il lavoro all'utente selezionato.
- ❑ **Send To Manager:** Spedisce il lavoro al manager dell'utente selezionato.
- ❑ **Send To Originator:** Spedisce il lavoro all'utente che ha iniziato il processo.
- ❑ **Send To Organizational Unit:** Spedisce il lavoro a tutti gli utenti dell'organizzazione(gruppo) selezionato.

5.3.1.5 Proprietà degli Eventi (*Events*)



Figura 5.3: Eventi di un'attività

Ogni attività può contenere uno o più eventi; È necessario decidere quali azioni o eventi sono necessari per completare l'attività. Nella figura di sopra l'evento è rappresentato dalla risposta del servizio informatico (se è presente o meno il materiale richiesto in scorta)

inserendo i dati dal form (interfaccia grafica). Da notare che l'ordine di esecuzione dell'evento è determinato dall'ordine in cui compare nell'interfaccia del blocco delle attività in k2.net Studio

The image shows a web form titled "Purchase Request". It contains several input fields: "First Name" with the value "Deon", "Last Name" with "Smith", "e-mail" with "deon@cygnusv.local", "Category" with a dropdown menu showing "Software", "Description" with "Workflow Weekly", "Motivation" with "Test", and "Price" with "230". Below these fields is a green section with the heading "Controllo del servizio informatico". Inside this section are two radio buttons: "Materiale disponibile" and "Materiale non disponibile". A yellow "Submit" button is positioned at the bottom right of the form.

Figura 5.4: Form dell'Evento nell'attività "controllo del servizio informatico"

a) Eventi predefiniti di K2.net

Client Event : Usato per facilitare l'interfazione tra gli utenti (attori) del workflow. Questo tipo di evento include :

- ⇒ **Client Side Code** – codice personalizzato dal programmatore.
- ⇒ **Internet** – un web-form che deve essere completato ed inviato da un utente.
- ⇒ **Mobile** – un web-form mobile form che deve essere completato ed inviato da un utente.
- ⇒ **SMTP Notification Message** – Un messaggio e-mail di notifica che può essere inviato ad un utente o qualsiasi indirizzo mail.
- ⇒ **Exchange** – un form di Outlook.

Server Event: codice personalizzato scritto dal programmatore che è eseguito lato server.

Server Mail Event : Usato per inviare e-mail ad un determinato indirizzo e-mail che contenere allegati.

IPC Server Event : Usato per comunicare con altri processi.

5.3.1.6 Linee di collegamento tra le attività (*Link the Activities with Line*)

In K2 net, le attività sono collegate tra loro con le linee(*lines*). In K2.net 2003 Studio ci sono sei diversi tipi di linee che hanno comunque le stesse funzioni.

Proprietà della linea (*Line Properties*)

- **Name:** Nome che descrive la linea . Esempio: materiale informatico.
- **Description:** testo libero per descrivere la linea.
- **Metadata:** Usato per caricare ulteriori dati non definiti come dati dei processi.
Usato principalmente quando questi dati sono manipolati al livello codice.
- **Line Rule:** una o più logica che può essere o vero(*True*) o falso(*False*). Se è vero il percorso rappresentato da questa linea sarà proseguita.

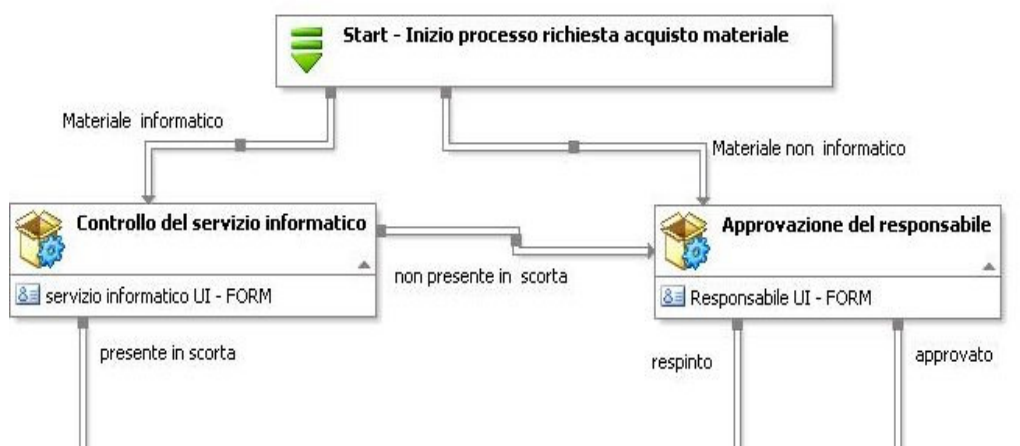


Figura 5.5: Linee di collegamento tra le attività

5.3.1.7 Campo dati del Processo (*Process Data Fields*)

Se noi pensiamo ad un processo in termini di informazione allora possiamo dire che un processo deve prima **raccogliere le informazioni rilevanti** che consentono di prendere le decisioni .

Uno dei metodi che possiamo usare per memorizzare le informazioni del processo è avvalersi di variabili di processo. In K2.net queste variabili di processo sono chiamate *Process Data Fields*.

I Vari tipi di dati disponibili sono: Boolean , Date, Decimal, Double , Integer,Long, String. Se necessario si può aggiungere il metadata (contiene le informazioni che descrive la variabile) per la variabile.

La variabile può essere nascosta.

Nota: I dati del processo e attività o i campi XML possono essere importati da un altro processo

5.4 Esportazione e test di un Processo

5.4.1 Esportazione

Dopo aver completato tutte le fasi di creazione di un processo (dalla concezione al disegno con K2.net Studio), bisogna adesso provare il processo. Di seguito ci sono i vari passaggi necessari per testare un processo:

1. **Esportare il processo** (*Export the process*) Il processo deve prima essere esportato da K2.net Studio. Durante l'esportazione del processo viene chiesto il server dove è stato installato il componente K2.net Server. Prima di essere esportato il processo viene compilato.
2. **Compilare l'applicazione ASP.NET** (*Compile and Build the ASP.NET application*): Da Visual Studio.NET, cliccare su "Build menu option".
3. **Dare i permessi di accesso al processo** (*Give users rights to the Process*): K2.net Service Manager è usato per dare i permessi agli utenti per i vari processi. Si può dare i permessi amministrare (*Admin*), avviare (*Start*), vedere (*View*).
4. **Testare il Processo.** Per meglio controllare il processo in esecuzione bisogna avviare la console di K2.net Server. Da questa console è possibile vedere i debug del processo.

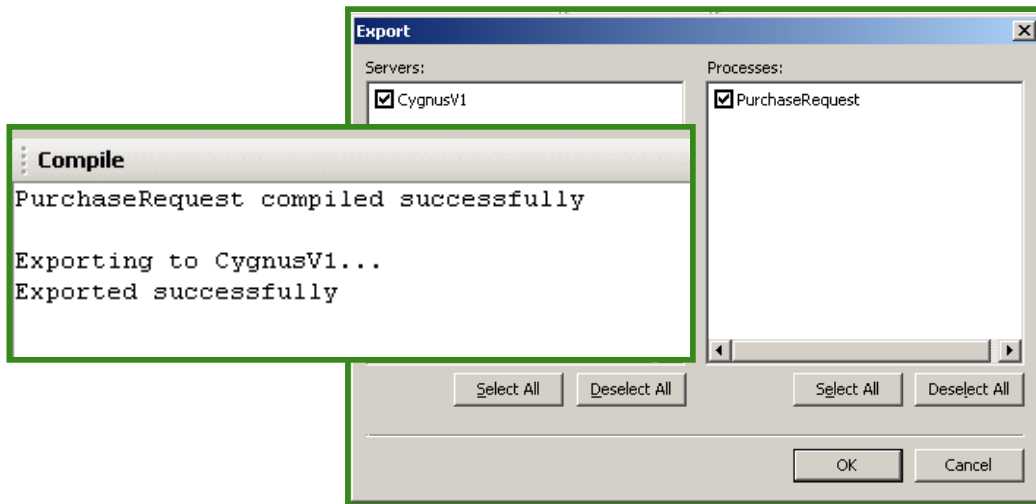


Figura 5.6: Compilazione ed esportazione del processo sul server

5.4.2 Istanza di processo

5.4.2.1 creazione della Istanze

Un processo workflow di solito inizia quando viene compilato un form via web (esempio richiesta acquisto nuovo hardware) ed inviato cliccando il pulsante del form. Fin dal momento in cui l'utente invia il form, K2.net server riceve un messaggio che ordina al server di pianificare e iniziare un nuovo processo. K2.net server esegue l'ordine creando una **nuova istanza di processo** (*Process Instance*).

Appena l'istanza del Processo arriva come attività, K2.net Server crea una nuova **istanza di attività**. Se l'attività ha varie eventi associati, il server dovrà creare una nuova **istanza di evento** per ogni evento.

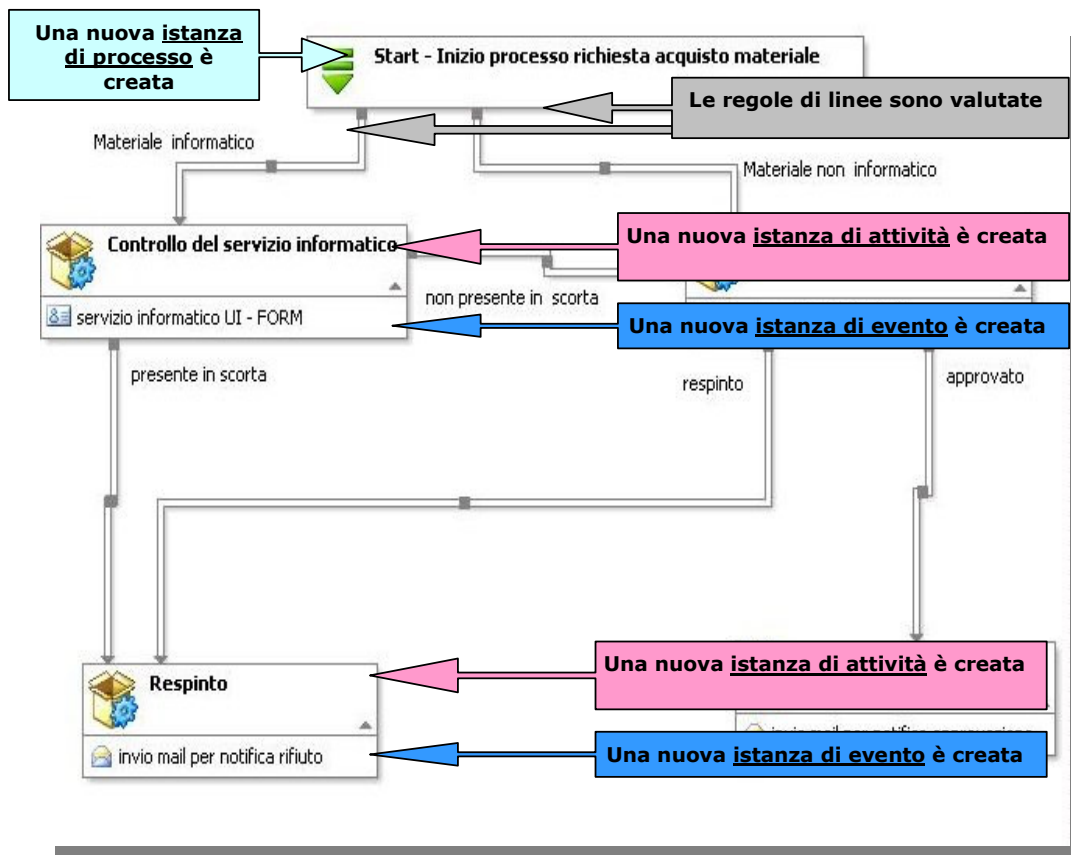


Figura 5.7: Istanze(Instance) create durante l'esecuzione di un processo

5.4.2.2 Proprietà delle Istanze

Durante l'esecuzione del processo, K2.net server preserva alcune proprietà per ogni istanza creata che sono elencati di seguito:

3.3 Istanza di processo (*Process Instance*)

- ❑ **Process Folio:** È il nome dato all'istanza di processo. È raccomandato ai programmatori assegnare un nome significativo a questa proprietà.
- ❑ **Start Date:** La data e ora di creazione dell'Istanza di Processo, in altre parole quando l'utente ha inviato il form iniziale.

- ❑ **Finish Date:** La data e ora in cui il processo arriva all'ultima attività e viene segnato come completato.
- ❑ **Originator:** Il nome dell'utente che ha avviato l'Istanza del Processo. In altre parole il nome dell'utente che ha inviato il form iniziale.
- ❑ **Status:** Lo stato corrente dell'Istanza del Processo. Lo stato può essere: **Attivo**(*Active*), **Completato**(*Completed*) e **Errore**(*Error*)
- ❑ **Duration:** Da quanto tempo il processo è eseguito o è stato eseguito in giorni, ore, minuti e secondi.

3.4 Istanza di attività (*Activity Instance*)

- ❑ **Activity Name:** Il nome dell'attività definito in K2.net Studio durante il disegno.
- ❑ **Start Date:** La data e ora di creazione dell'Istanza di Attività.
- ❑ **Finish Date:** La data e ora in cui l'Istanza di Attività è stato impostato come Completato.
- ❑ **Priority:** È la Priorità dell'Attività definita in K2.net Studio. I valori possono essere: High, Medium e Low.
- ❑ **Status:** Lo stato corrente dell'Istanza di attività. Lo stato può essere: **Attivo**(*Active*), **Completato**(*Completed*) e **Errore**(*Error*).
- ❑ **Duration:** Da quanto tempo il processo è, o è stato, attivo in giorni, ore, minuti e secondi.

3.5 Istanza di Evento(*Event Instance*)

- ❑ **Event Name:** Il nome dell'Evento definito in K2.net Studio durante il disegno.
- ❑ **Start Date:** La data e ora di creazione dell'Istanza di Evento.
- ❑ **Finish Date:** La data e ora in cui l'Evento è stato impostato come Completato.
- ❑ **Destination:** Indica il responsabile che può eseguire l'azione, definito da "Activity's Destination Rule" in K2.net Studio.
- ❑ **Priority:** È la Priorità dell'Evento definita in K2.net Studio. I valori possono essere: High, Medium e Low.
- ❑ **Status:** Lo stato corrente dell'Istanza di Evento. Lo stato può essere: **Attivo**(*Active*), **Completato**(*Completed*) e **Errore**(*Error*)
- ❑ **Duration:** Da quanto tempo l'Evento è, o è stato, attivo in giorni, ore, minuti e secondi.

6. K2.NET WORKSPACE E SERVICE MANAGER

6.1 K2.net Workspace

K2.net Workspace offre un'interfaccia utente basata sul web e personalizzabile che può essere usata per ordinare dinamicamente i dati di flusso di lavoro e dati di rapporto di filtro. È possibile avere i vari rapporti (**Report**) grafici sui processi, attività..., fare ricerca (**Search**) e vedere la lista di lavoro (**Worklist**) per ogni utente.

Gli indirizzi di accesso dei componenti di K2.net Workspace sono :

Report: <http://server/k2v3/workspace/Report.aspx> (Rapporti grafici dei processi...)

Search: <http://server/k2v3/workspace/Search.aspx> (ricerca di attività...)

Worklist: <http://server/k2v3/workspace/WorkList.aspx> (lista di lavoro di ogni utente)

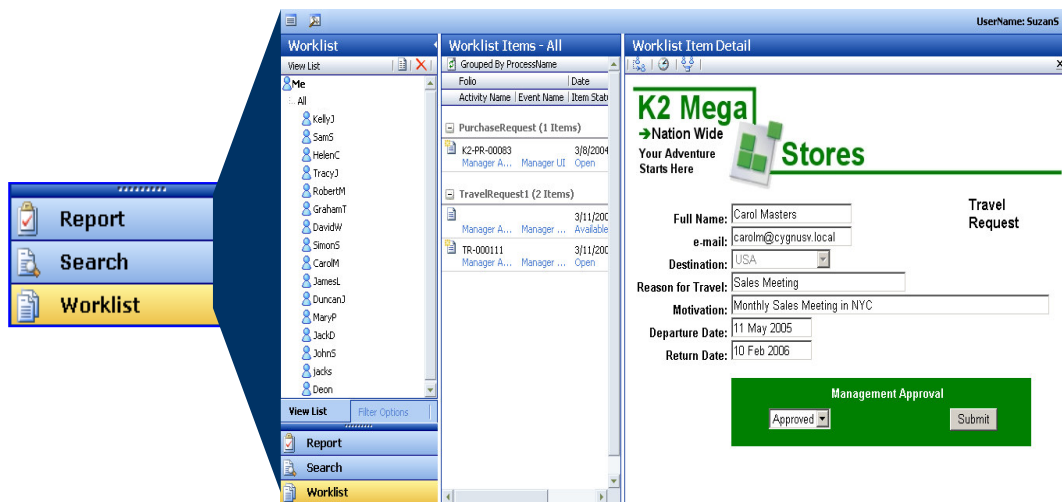


Figura 6.1: K2.net Workspace : interfaccia grafica

6.2 K2.net Service Manager

È uno dei componenti più importante e potenti di K2.net 2003. Esso permette di :

- ⇒ Gestire K2.net Servers (anche vari server in *cluster* come in singolo server logico) sia in locale che in remoto (da internet).
- ⇒ Gestire i Processi.

- ⇒ Gestire gli Errori.
- ⇒ Gestire utenti e Gruppi.
- ⇒ Gestire “Destination Queues” .
- ⇒ Gestire Worklists(liste di lavoro).
- ⇒ Gestire la sicurezza.

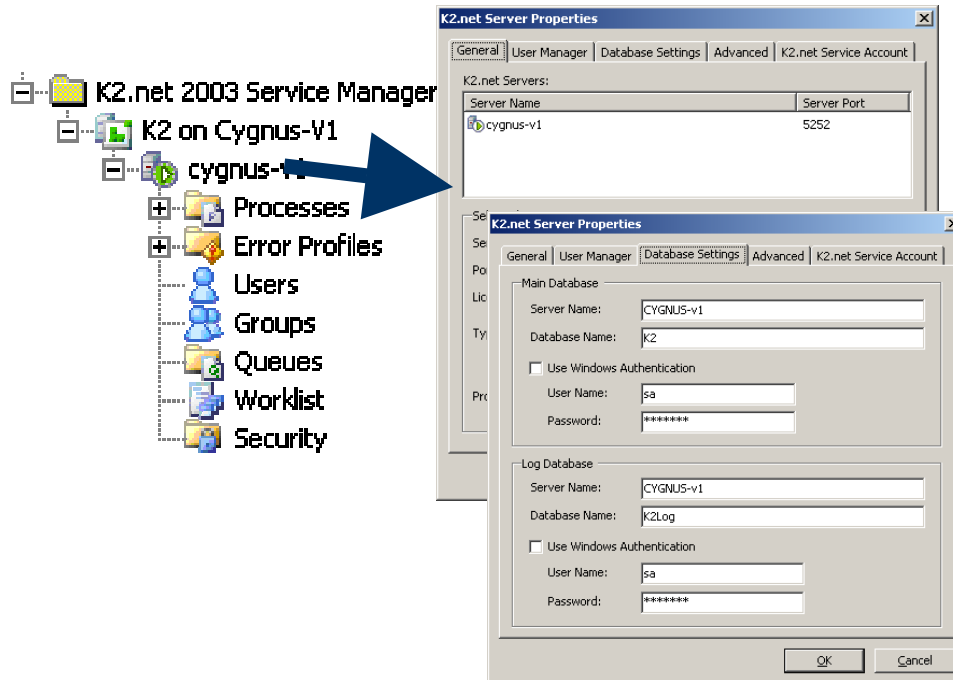


Figura 6.2: K2.net 2003 Service Manager

6.2.1 Gestione dei Processi (*Managing Processes.*)

Nota: Solo le istanze di processi in esecuzione possono essere gestiti o manipolati. Un Processo una volta Completato, non può più essere soggetto ad azioni nel “System Manager”

Dall’interfaccia di K2.net 2003 Service Manager è possibile gestire i processi attraverso :

Le versioni (*Versions*) dei processi

- Cambiare la versione predefinita.(La nuova versione è generata ogni volta che il processo è esportato da K2.net Studio. L’ultima versione esportata diventa sempre la versione predefinita da cui vengono avviate le nuove Istanze di Processi.)

Le Istanze (*Instances*) dei processi

- ❑ Vedere le Istanze in esecuzione.
- ❑ Fermare le Istanze.
- ❑ Cancellare le Istanze.
- ❑ Reindirizzare ad altri Attività.

Utenti e Gruppi (*Users and Groups*)

⇒ Assegnare I permessi di amministrare, avviare vedere agli utenti.

7. ANALISI DELLE REGOLE DELLE ATTIVITÀ DI UN PROCESSO

7.1 Introduzione

Una Attività all'interno di un Processo può contenere potenzialmente Eventi di utenti (ricevuti e gestiti dall'uomo: esempio un form da compilare), Eventi di Sistema (ricevuti e gestiti dalla macchina: esempio invio di una mail), e eventi di intercomunicazione tra Processi (IPC= *Inter Process Communication*). Di solito la grande parte delle **logiche d'impresa** (*business logic*) sono contenute in un'Attività.

7.2 Le logiche d'impresa(*business logic*)

Le logiche d'impresa(*business logic*) includono questi concetti:

- Chi sarà coinvolto nell'Attività?
- Ci sono pre-requisiti prima che l'utente possa avviare un'Attività?

L'Attività è la componente più importante del Processo perché è il punto in cui le decisioni sono prese, i dati sono forniti e le azioni sono prese. Guardando le proprietà di un'Attività ci si accorge subito che la maggior parte sono delle regole (*rules*).

7.3 Regole d'Attività(*Activity Rule*)

Possiamo dire che le regole **Regole d'Attività** (*Activity Rule*), rappresentano lo strumento usato per **costruire le logiche d'impresa** (*business logic*) nell'Attività.

Nella figura successiva sono illustrate le varie regole gestite dall'Attività di un Processo

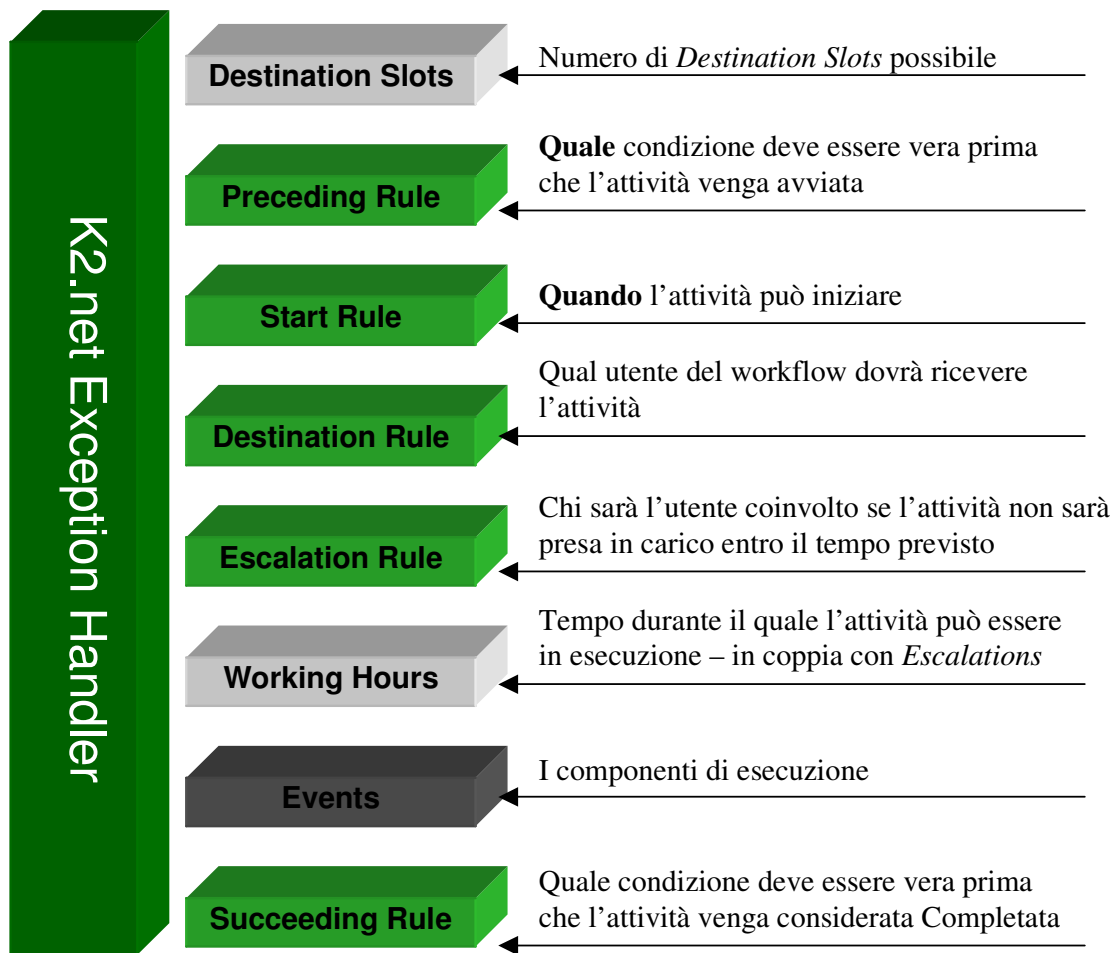


Figura 7.1: Varie regole di un Attività

Vediamo in dettaglio alcune regole molto importanti nei paragrafi successivi.

7.3.1 Preceding Rule

Se vogliamo posticipare l'avvio di un'Attività, non nel tempo (in termini di minuti, ore o giorni), ma finché certe **condizioni si siano verificate**, allora bisogna definire un **Preceding Rule**. In altre parole il *Preceding Rule* indica sotto quale condizione una nuova istanza di Processo viene creata.

Esempio: Un Processo (vedi figura di sotto) diviso in due Attività parallele A e B e un incontro (*rendez-vous*) all'Attività C. E vogliamo che le due Attività A e B siano completate prima che l'Attività C inizi. Allora bisogna creare il *Preceding Rule* per l'Attività C.

Nel modello della WFMC questo tipo di scomposizione del Processo viene chiamato *AND-JOIN*.

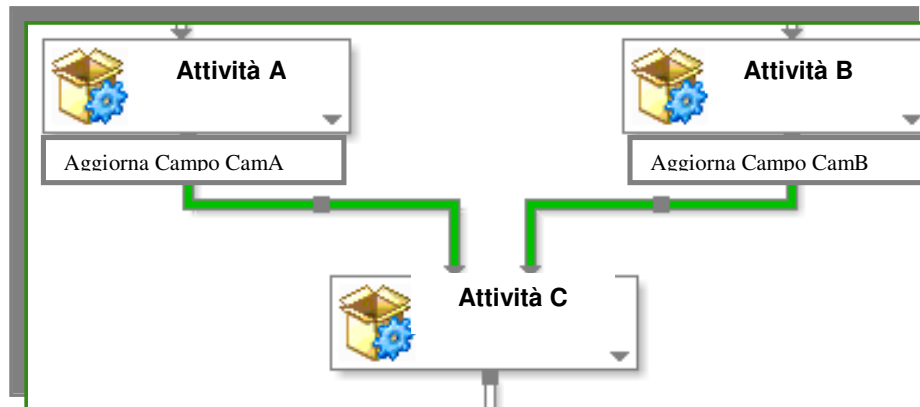


Figura 7.2: Esempio di applicazione di Preceding Rule per l'Attività C

Nell'esempio della figura precedente l'utente di ogni attività (utente Attività A e Utente Attività B) dovrà abilitare, dalla sua lista di lavoro (*Worklist*), i vari campi **camA** e **camB** Vero (*True*) o Falso (*False*) come illustrato nella seguente figura.

La validazione (del *Preceding Rule*) dei campi prima di avviare l'attività C è :

K2.Process.Datafields("camA")=True And K2.Process.Datafields("camB")= True

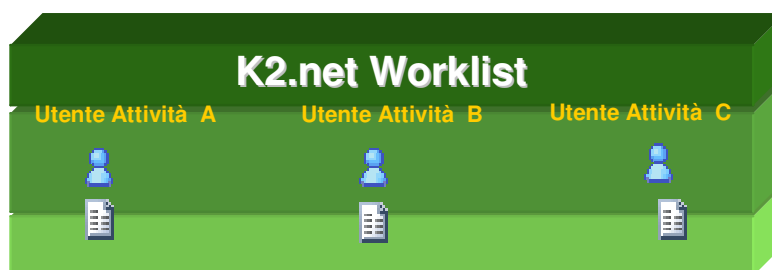


Figura 7.3: Lista di lavoro(Worklist) degli utenti per ogni attività

7.3.2 Destination Rule e Activity Slots

Se si specifica che un lavoro deve essere spedito a **più di un utente**, nella regola di destinazione (*Destination Rule*), dell'attività allora è necessario chiedersi **quanti di questi**

utenti di destinazione sono richiesti ad attivare il lavoro spedito prima che si completi l'attività.

Per esempio nello scenario dell' attività di Help-Desk (Assistenza alla Clientela), dove i lavori (richieste di assistenza) sono inviati a tutti gli utenti del gruppo Help-Desk. Il primo utente del gruppo che apre questa richiesta di assistenza (lavoro), di fatto lo prende in carico e diventa, di conseguenza, il responsabile di quella richiesta. Appena un utente ha preso in carico la richiesta, questa (richiesta di assistenza) è automaticamente rimossa dalla lista di lavoro di tutti gli altri utenti del gruppo Help-desk. Qui sorge un dubbio; Abbiamo bisogno che un'utente solo prenda in carico il lavoro(richiesta del cliente) togliendolo dalla lista di lavoro degli altri ? Per rispondere a questa domanda basta impostare il parametro “**Activity Slots**”. Per quest'attività è necessario un solo “Activity Slots”

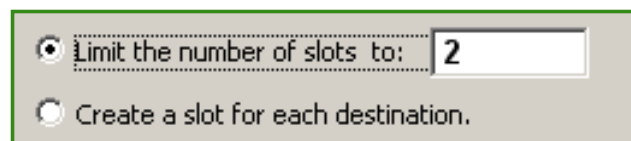
In sintesi il numero di “Activity Slots” definito per un'Attività determina quanti utenti workflow possono **prendere in carico lo stesso lavoro simultaneamente**. Lo stato iniziale del lavoro messo nella lista di lavoro(*Worklist*) è disponibile(*Available*) , quando un utente **apre** e un “**Activity Slot**” vuoto è preso. Ma appena **l'ultimo “Activity Slot” è preso** allora tutti i lavori con stato disponibile sono rimossi dalla lista di lavoro di quei utenti. Di conseguenza specificando **più di un “Activity Slot”**, vuole dire che siamo interessati ad avere una **risposta da ogni singolo utente**.

Esempio:

Inviare un lavoro a un gruppo di 5 utenti (persone) ma solo i primi due che lo apriranno ci potranno lavorare.

In questo caso bisogna: Impostare il numero di “Activity Slot” a 2

Appena i primi due utenti apriranno il lavoro nella loro lista di lavoro, sarà allocato per quei due utenti e sparirà dalla lista di lavoro degli altri tre utenti.



The image shows a configuration dialog box with a light gray background and a green border. It contains two radio button options. The first option is selected, indicated by a black dot inside the circle. The text next to it is "Limit the number of slots to:" followed by a text input field containing the number "2". The second option is unselected, indicated by an empty circle, and its text is "Create a slot for each destination."

Figura 7.4: Impostazione del numero di “Activity Slot”

7.3.3 Succeeding Rule e Activity Slots

Questa regola definisce sotto quale condizione un'Attività può essere considerata completata.

Ha senso usare "Succeeding rule" solo con vari "Activity Slots" perché :

- Un'Attività viene definita completata se la **condizione definita dall'utente** e vera **oppure tutte liste di lavoro sono completate.**
- Per 1 utente di destinazione il "Succeeding rule" viene impostato a vero appena quel utente ha completato il suo lavoro indipendentemente dalla condizione definita

In sintesi se si ha:

⇒ Numero di Slots = 1 => non è richiesto il "Succeeding Rule"

⇒ Numero di Slots > 1 => è richiesto il "Succeeding Rule"

7.4 Gestione delle eccezioni e errori

Durante l'esecuzione della Attività di un processo, eccezioni o errori possono succedere.

E possibile scrivere un codice per esempio in C# per gestire questi eventi per :

⇒ inviare SMS o e-mail al gruppo di programmatori (o gruppo di gestione del progetto)per avvisare dell'errore

⇒ Prova di risolvere il problema e fare proseguire il processo

Esempio di codice scritto in C#

```
public void Main(ExceptionContext K2)
{
    System.Web.Mail.MailMessage objMsg = new System.Web.Mail.MailMessage();
    object strFrom = "";
    object strEmail = "";
    object strSubject = "";
    object strBody = "";
    System.Xml.XmlDocument oXMLDoc = new System.Xml.XmlDocument();
    System.Xml.XmlNode oNode = null;
    System.Web.Mail.SmtpMail.SmtpServer =
    SourceCode.K2Utilities.GeneralMod.GetDefaultSMTPServer();
    strFrom = "k2server@cygnusv.local";
    strEmail = "deon@cygnusv.local";
    strSubject = "An Error has occurred";
    strBody = "Error Date: " + System.DateTime.Now.ToString +
```

```
System.Environment.NewLine +  
"Error Process: " + K2.ProcessInstance.Process.Name +  
System.Environment.NewLine +  
"Error Item Name: " + K2.ContextType.ToString + System.Environment.NewLine;  
objMsg.From = strFrom.ToString();  
objMsg.To = strEmail.ToString();  
objMsg.Subject = strSubject.ToString();  
objMsg.Body = strBody.ToString();  
System.Web.Mail.SmtpMail.Send(objMsg);  
End Sub
```

8. K2.NET RUNTIME OBJECT MODEL

8.1 Introduzione

Questo capitolo molto importante permette di capire come fa interagire un software(applicazione) esterno con un processo di K2.net. Prima di tutto bisogna vedere come è strutturato questo componente. K2ROM è la sigla di **K2.net Runtime Object Model** (E un insieme completo di **API esposti come .NET assemblies e WSDL Web Services**).

8.2 Definizione

In generale K2ROM può essere definito come insieme di strumenti che possono essere **utilizzati per accedere ed interagire con i processi in esecuzione sul K2.net server**. Ogni processo corrente è rappresentato come un'Istanza di Processo, in questo modo possiamo dire K2ROM ci fornisce le interfacce necessarie per **facilitare l'esposizione e la gestione degli Istanze di Processo** che sono in esecuzione sul server. quello sta funzionando sull'assistente. Possiamo anche esporre naturalmente e gestire tutti i componenti che compongono un'Istanza di Processo come Istanza di Attività e di evento e così via.

8.3 Strumenti forniti dal K2ROM

L'insieme degli strumenti forniti dal K2ROM **include** sia l' *assembly .NET* chiamato **K2ROM.dll** ed anche i **Web services**, il servizio di Worklist(lista di lavori) e il FilterService. In **questo capitolo** ci concentriamo sul l' *assembly .NET* che espone il K2ROM come *namespace* con un insieme di oggetti definiti. Ecco di seguito i vari oggetti che saranno descritti in questo capitolo :

- Gli oggetti per connessione(*The Connection Object*).
- Gli oggetti per Istanze di Processi(*The ProcessInstance Object*).
- Gli oggetti per Istanze d'Attività (*The ActivityInstanceDestination Object*).
- Gli oggetti per liste di lavoro(*The Worklist Object*).
- Gli oggetti per gli elementi della lista di lavoro(*The Worklist Item Object*).

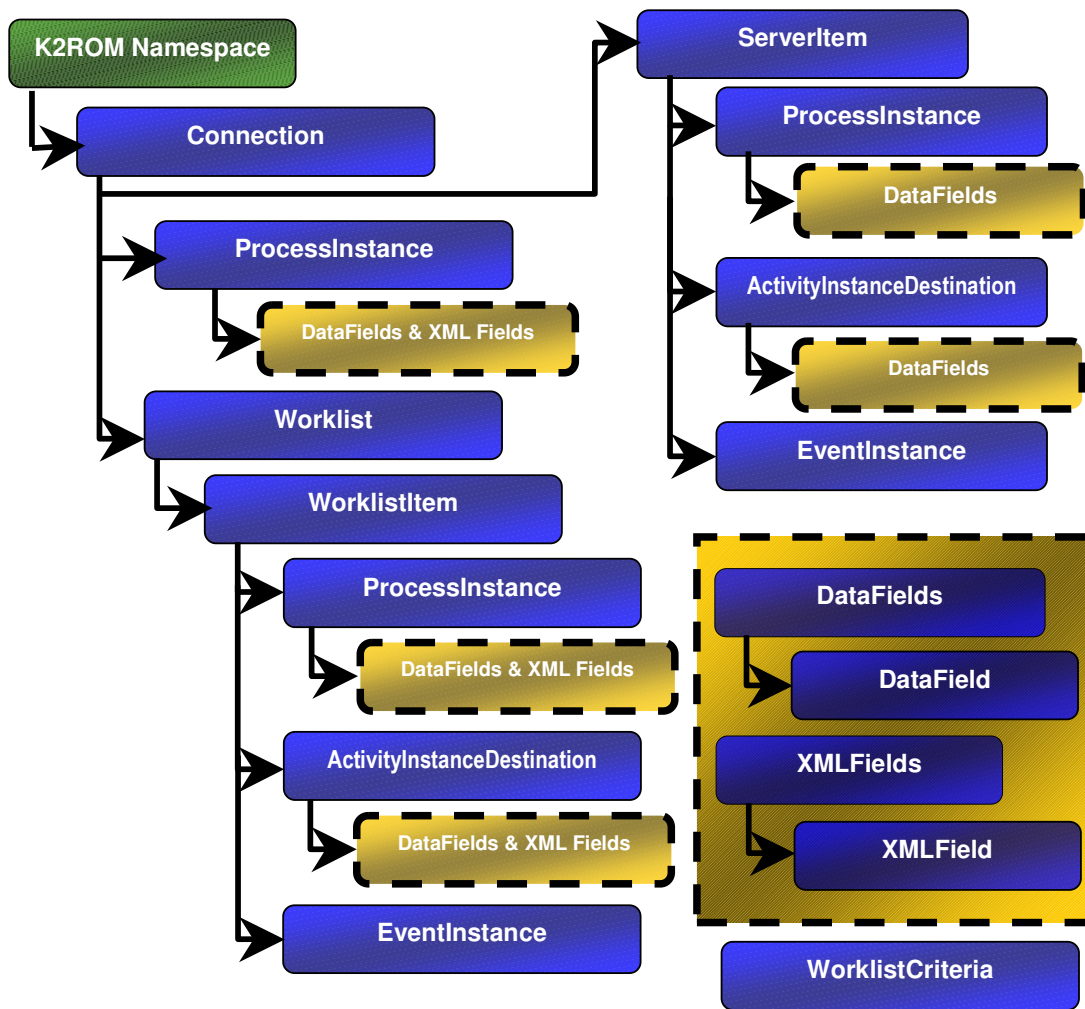


Figura 8.1: Gli Oggetti di K2ROM

8.4 Gli oggetti di K2ROM (*K2ROM Objects*)

8.4.1 Introduzione

L'assembly K2ROM.DLL è esposta come il *namespace* di K2ROM che consiste in un insieme di undici oggetti. Ogni oggetto è implementato come una classe pubblica con i

metodi e le proprietà. Vedremo come ogni oggetto è Istanziato, i metodi con i loro parametri ed anche le proprietà ogni oggetto.

Uso della dll :

Prima di usare l'assembly K2ROM è necessario creare una referenza ad esso da Visual Studio. NET Project. La DLL di K2ROM(K2ROM.DLL) si trova nella cartella "bin" di installazione di K2.net 2003 Studio.

8.4.2 Gli oggetti per la connessione(The Connection Object)

8.4.2.1 Introduzione

Di tutti gli oggetti nel K2ROM, l'oggetto di connessione è l'unico oggetto che è manualmente istanziato una **nuova** parola chiave. Tutti gli altri oggetti sono Instantiated chiamando un metodo o una proprietà di quel oggetto. L'oggetto di connessione è la chiave per usare il modello Runtime dell'oggetto (Runtime Object Model); nessun azione può essere eseguita su qualunque Istanza di Processo prima che una connessione a K2.net Server sia stata aperta.

Dopo che un oggetto di connessione è stato istanziato, una connessione al server è effettuata chiamando il metodo. Il metodo aperto può essere chiamato con o senza il nome dell'utente. Se il nome dell'utente non è specificato, le credenziali dell'utente che attualmente è impostato in IIS saranno usate per stabilire il collegamento. È molto importante notare che una volta aperta la connessione con il server, **tutto sarà fatto nel contesto dell'utente che ha aperto il collegamento**, per esempio: Se desiderate accedere ad un Worklist, il Worklist dell'utente che ha aperto il collegamento sarà restituito. In questo capitolo ci riferiamo all'utente che è stato usato dallo sviluppatore per aprire il collegamento come '**Utente di connessione**'.

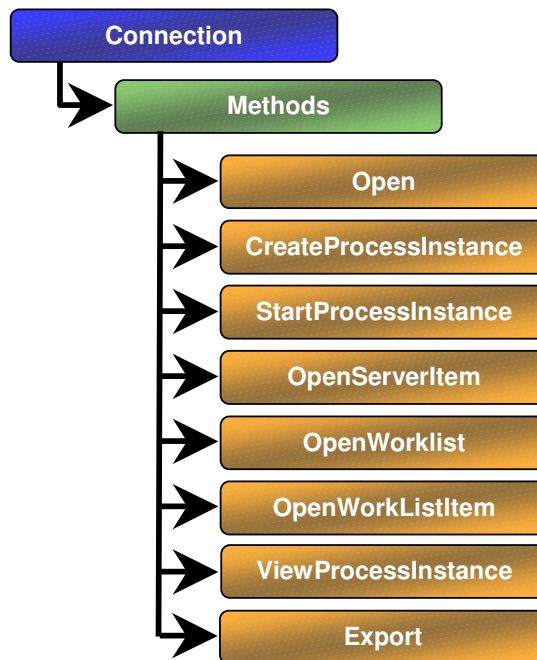


Figura 8.2: I vari metodi dell'oggetto "Connection"

8.4.2.2 Parametri di connessione

Nota: I parametri della stringa di connessione dell'**apertura del metodo(Open Method)** possono essere usati nei seguenti modi:

Connessioni per le versioni precedenti a *K2.net 2003 Server SP1*

`strCon = Domain,User,Password`

Connessioni per le versioni successive a *K2.net 2003 Server SP1*:

`strCon = [;];Authentication=Windows;Domain=[Domain];User=[UserName];Password=[Password]`

`strCon = [;];Authentication=External;[Your Custom Constring]`

`strCon = [;];SecurityPackage=NTLM;Domain=[Domain];User=[UserName];Password=[Password]`

`strCon = [;];SecurityPackage=Kerberos;Domain=[Domain];User=[UserName];Password=[Password]`

`strCon = Domain,User,Password`

Esempio di codice in C#.NET per accedere ai metodi dell'oggetto "Connection"

Nome della classe (Class Name)	Connessione(Connection)
Constructor	<pre>public Connection()</pre> <p>Esempio: K2ROM.Connection myConn = new K2ROM.Connection();</p>
Open	<pre>public void Open (string Server);</pre> <pre>public void Open (string Server ,string ConStr);</pre> <p>Esempio myConn.Open("k2megasrv"); oppure myConn.Open("k2megasrv", "k2mega,SuzanS,k2pass");</p>

8.4.2.3 Metodi di Connessione : Creazione/avvio Istanze di Processi

I metodi *CreateProcessInstance* (per creare le istanze di processi) con *StartProcessInstance* (per avviare le istanze di processi) sono usati se desidera avviare o progettare(creare) un nuovo Processo sul K2.net Server. Il metodo di *CreateProcessInstance* restituisce un riferimento ad un nuovo *ProcessInstance*. Questa nuova Istanza può allora essere manipolato(esempio le impostazioni dei dati del Processo, le priorità ecc...)

Dopo che i valori necessari sono stati impostati, si può progettare l'Istanza del Processo chiamando il metodo di *StartProcessInstance*.

Esempio di Codice in C#

Nome della classe (Class Name)	Connessione(Connection)
CreateProcessInstance	<pre>public K2ROM.ProcessInstance CreateProcessInstance(string ProcName);</pre> <pre>public K2ROM.ProcessInstance CreateProcessInstance(string ProcName,int Version);</pre> <p>Esempio: K2ROM.ProcessInstance MioProcess = myConn.CreateProcessInstance("Purchasing\PR"); Dove Purchasing è il nome del progetto e PR il nome del processo</p>

StartProcessInstance	<pre>public void StartProcessInstance(K2ROM.ProcessInstance ProcInst);</pre> <pre>public void StartProcessInstance(K2ROM.ProcessInstance ProcInst,bool Sync);</pre> <p>Esempio: myConn.StartProcessInstance(MioProcess);</p>
-----------------------------	---

8.4.2.4 Metodi di Connessione : Apertura eventi sul server (“OpenServerItem”)

Prima di tutto bisogna ricordare che un evento server può essere sincrono o asincrono.

Un **Evento Server Sincrono**(*Synchronous Server Event*) esegue tutto il codice nel blocchetto di codice e poi si **complete**. Dopo che è stata completata l’Istanza di quell’evento Server non esiste più anche se salviamo il Numero Seriale dell’Evento, si riceverà un errore se si tenta di accedere all’Evento Server usando quel Numero Seriale.

Un **Evento Server Asincrono**(*Asynchronous Server Event*) invece non Complete finche l’ultimo metodo dell’evento sia stata chiamata(in modo programmato). L’unico modo in cui il server può essere accessibile e specificare il suo unico Numero Seriale(**Serial Number**)

Esempio di Codice in C#

Nome della classe (Class Name)	Connessione(<i>Connection</i>)
OpenServerItem	<pre>public K2ROM.ServerItem OpenServerItem(string SerialNumber);</pre> <p>Esempio: K2ROM.ServerItem myServItem = myConn.OpenServerItem("K2MEGASRV,11,11");</p>

8.4.2.5 Metodi di Connessione : Accesso Worklist (Lista di lavoro)

L'unico modo per **accedere ad Un’Istanza di Processo**, è attraverso l’elemento Worklist(Lista di lavoro) nella Worklist del partecipante(utente) di workflow. Il K2ROM

fornisce due metodi per accedere agli elementi della Worklist oppure ad un specifico elemento della Worklist. Questi due metodi funzionano nel contesto dell'utente che è stato usato per aprire il collegamento al K2.net Server. il metodo **OpenWorkList** restituirà un oggetto che è una collezione di elementi del Worklist.

Esempio : **myWorkList = myK2Conn.OpenWorkList("ASP")**

Ogni elemento nella Worklist può essere controllato e manipolato enumerando la collezione degli elementi, ogni elemento inoltre è rappresentato da un oggetto, chiamato l'oggetto **WorkListItem**.

Esempio : **MyWorkListItem = myWorList.Item(i)**

Il metodo **ViewProcessInstance** può essere usato per ottenere le informazioni sullo stato attuale di una specifica Istanza di Processo. L'Istanza di Processo è specificato passando l'ID dell'Istanza di Processo come parametro nella chiamata di metodo. Il metodo restituisce una stringa in formato XML che può essere usato per ricostruire una tabella di ViewFlow dell'Istanza di Processo.

Esempio : **myXMLString= myK2Conn.ViewProcessInstance("ProcessInstanceID")**

Nota: Il metodo di ViewProcessInstance può essere chiamato soltanto se l'utente di connessione è l'Amministratore (*Administrator*) del processo, o se l'utente collegato ha un elemento Worklist da quest'Istanza di Processo nella sua Worklist o nella Worklist dell'amministratore degli utenti.

Esempio di Codice in C# (Gestione della Worklist)

Nome della classe (Class Name)	Connessione(Connection)
OpenWorkList	<pre>public K2ROM.Worklist OpenWorklist(string Platform); public K2ROM.Worklist OpenWorklist(string Platform,string ManagedUser); public K2ROM.Worklist OpenWorklist(WorklistCriteria Criteria); Esempioample K2ROM.Worklist myWList =myConn.OpenWorklist("ASP", "KellyJ");</pre>
OpenWorkListItem	public K2ROM.WorklistItem OpenWorkListItem (string

	<pre> SerialNumber,string Platform); public K2ROM.WorklistItem OpenWorkListItem(string SerialNumber,string Platform,bool Alloc); public K2ROM.WorklistItem OpenWorkListItem(string SerialNumber, string Platform, bool Alloc,bool IgnoreStatus); Esempio K2ROM.WorklistItem myLItem = myConn.OpenWorkListItem("K2SRV,1,1","ASP"); </pre>
ViewProcessInstance	<pre> public string ViewProcessInstance(int PID); Esempio: string myProcData =myConn.ViewProcessInstance(127); </pre>

8.4.2.6 Metodi di Connessione : Proprietà pubblica dell'Oggetto *UserList*

L'Oggetto di connessione ha una proprietà pubblica, che è un oggetto chiamato *User (Utente)*. Questo oggetto può essere *usato per recuperare informazioni supplementari sull'utente che ha aperto la connessione al K2.net Server*. Le proprietà che possono essere recuperate includono il **nome**, **e-mail**, l'**indirizzo**, l'amministratore (*Manager*), che restituisce l' Oggetto *UserList*.

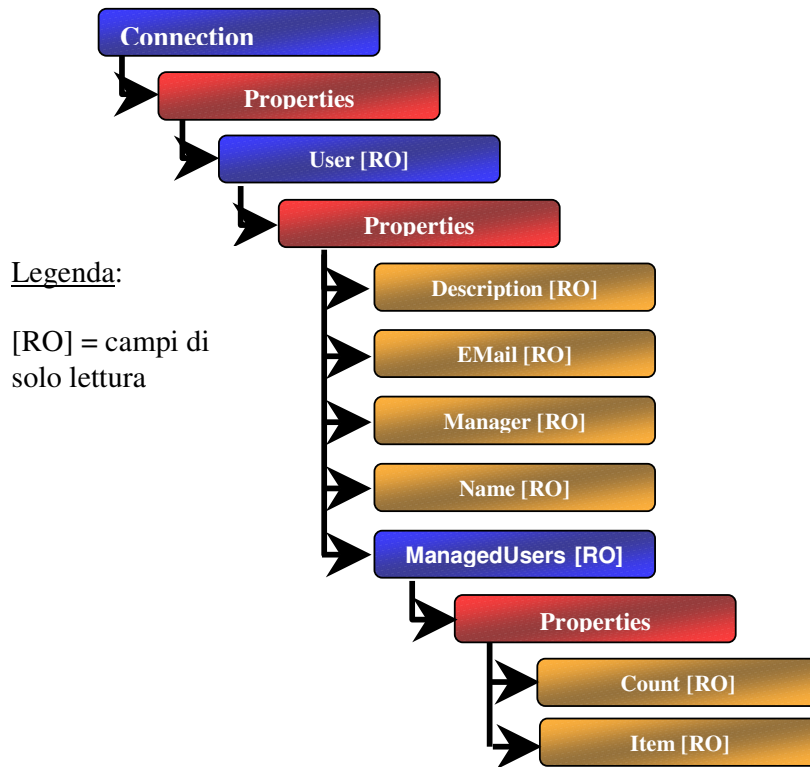


Figura 8.3: Proprietà dell' Oggetto Utente(User Object) – Utente usato per stabilire la connessione con K2.net Server (Connection User)

Definizione dell' oggetto in C#.NET

Nome della classe (Class Name)	Connessione(<i>Connection</i>)
User	<pre> public K2ROM.Worklist OpenWorklist(string Platform); public K2ROM.Worklist OpenWorklist(string Platform,string ManagedUser); public K2ROM.Worklist OpenWorklist(WorklistCriteria Criteria); Esempio K2ROM.Worklist myWList =myConn.OpenWorklist("ASP","KellyJ"); </pre>

8.4.2.7 Metodi di Connessione : Filtro e ordinamento della Worklist con l'oggetto *WorklistCriteria*

Questo oggetto è usato per creare i criteri di filtro e ordinamento che possono essere usati insieme al metodo di *OpenWorklist*. Il metodo di ***AddSortField*** permette di aggiungere un determinato filtro ordinato (crescente o decrescente) su determinati campi di Istanze di Processo. Esempio: *ProcessFolio*, *ActivityPriority*, *ActivityStartDate*, *ProcessStartDate*, *ProcessPriority*. Il metodo di ***AddFilterField*** permette di creare varie combinazioni di filtri basati sui campi di Istanze di Processo. Esempio : *ProcessFolio*, *ActivityPriority*, *ActivityStartDate*, *ProcessStartDate*, *ProcessPriority*.

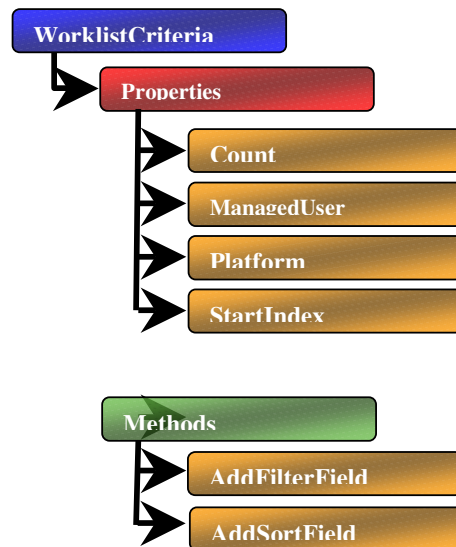


Figura 8.4: Proprietà dell'oggetto *WorklistCriteria* e vari metodi

Definizione dell'oggetto in C#.NET

Nome della classe (Class Name)	WorklistCriteria
Count	public int Count {get; set;} Esempio myWLCriteria.Count = 100;
ManagedUser	public string ManagedUser {get; set;} Esempio myWLCriteria.ManagedUser = "KellyJ";

Platform	public string Platform {get; set;} Esempio myWLCriteria.Platform = "ASP";
StartIndex	public int StartIndex {get; set;} Esempio myWLCriteria.StartIndex = 51;

Definizione dei metodi dell'Oggetto *WorklistCriteria* in C#.NET

Nome della classe (Class Name)	WorklistCriteria
AddFilterField	public void AddFilterField (WCField Field,WCCompare Compare, object Value); public void AddFilterField (WCLogical Logical,WCField Field, string SubField,WCCompare Compare,object Value); public void AddFilterField (WCLogical Logical,WCField Field, WCCompare Compare,object Value); Esempio myWLCriteria.AddFilterField(K2ROM.WCLogical.And, _ K2ROM.WCField.ProcessData, _ "Department",K2ROM.WCCompare.Equal, "HR");
AddSortField	public void AddSortField (WCField Field,string SubField, WCSortOrder SortOrder); public void AddSortField (WCField Field,WCSortOrder SortOrder); Esempio myWLCriteria.AddSortField(K2ROM.WCField.ProcessData, _ "Department", K2ROM.WCSortOrder.Ascending)

Nota: Il metodo **AddFilterField** non consente il raggruppamento delle espressioni logiche. E di competenza del programmatore espandere le espressioni logiche.

Esempio:

Vogliamo selezione tutti i lavori dove i dipartimenti (Department) hanno il nome che contiene le lettere "HR" e che abbiano Priorità Alta o Media

A = Department like "HR"

B = Priority = "HIGH"

C = Priority = "MEDIUM"

espressione logica: Risultato = A AND (B OR C)

espressione espansa: Risultato = A AND B OR Risultato = A AND C

8.4.2.8 L'oggetto Istanza di Processo (*Process Instance Object*)

L'unico modo in cui un Istanza di Processo, insieme alle Istanze di Attività e Istanze di Evento associate, possono essere accessibili è attraverso un specifico elemento della Worklist (lista di lavoro) dell'utente di Workflow, oppure nel caso di elemento di evento di server (Event Server), aprendo l'elemento dell'Istanza di Eventi di Server con il metodo *OpenServerItem* passando l'unico numero seriale di Evento (*Event Serial Number*) come parametro

Una volta ottenuto un riferimento nella Worklist o l'elemento server, è possibile accedere alle proprietà di *ProcessInstance* dell'oggetto e **manipolare** il *DataFields*, i campi di *XML*, *Priority* (priorità), *Expected Duration* e *Folio* dell'Istanza di Processo. La maggioranza delle proprietà dell'Istanza di Processo sono di **solo lettura** (*Read Only*) e non possono essere impostati. Queste proprietà includono *StartDate*, *Name* (nome), *Description* (descrizione) e *Metadata* (meta dati).

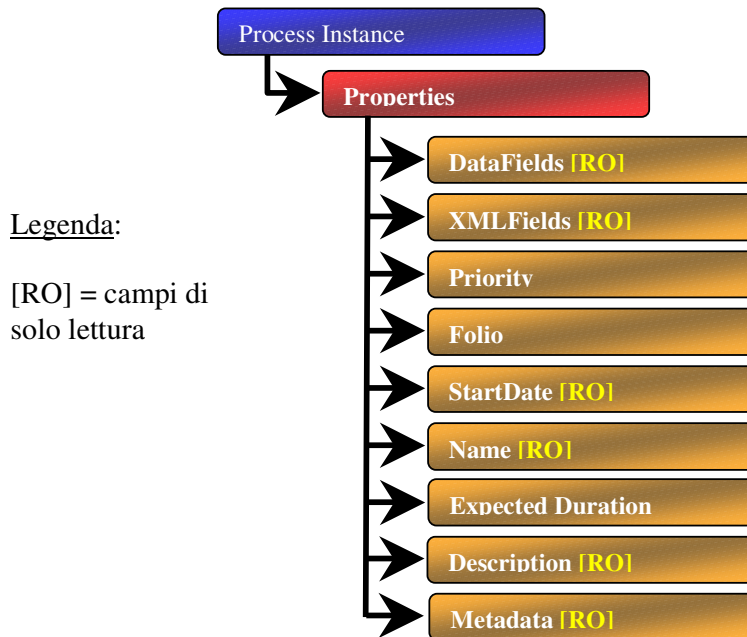


Figura 8.5: Le proprietà dell'Istanza del processo

8.4.2.9 Proprietà dell'Oggetto Process Instance

a) Proprietà del *DataFields*(campo data)

La proprietà di *DataFields* dell'oggetto Istanza di processo è di sola lettura. Ciò non significa che i valori del campo d'informazione non possono essere cambiati. Significa che una collezione di *DataFields* non può essere modificata aggiungendo o rimuovendo *DataFields* in modo programmato. **La definizione reale di un *DataField*, come definita in K2.net Studio non può anche essere cambiato usando il K2ROM.**

E possibile in ogni caso cambiare il valore del *DataField* accedendo attraverso la proprietà dell'elemento di Collezione dei *DataFields* e modificando il valore della proprietà.

Si noti che la proprietà dell'elemento dell'oggetto *DataFields* è impostata come la proprietà predefinita, che significa che non è necessario specificare esplicitamente la proprietà.

myValue = myProcInst.DataFields.Item("FieldName").Value
myValue = myProcInst.DataFields("FieldName").Value

b) Proprietà : XML Fields campi di XML

I campi di XML sono manipolati nello stesso modo che di un normale *DataFields*, anche se il contenuto di questi campi è notevolmente differente . Rappresenta la collezione di campi XML per questo Processo.

Ogni campo XML è accessibile attraverso la proprietà dell'elemento dell'Oggetto XMLFields.

c) Proprietà: Folio

La proprietà **Folio** dell'Istanza di Processo è spesso trascurata. Il Folio fornisce un unico identificatore per ogni Istanza di Processo guardando una lista di Istanza di Processo dal Workspace. Quindi è consigliabile creare uno schema per la generazione dei propri nomi di Folio.

Come impostare o avere il nome comune dell'istanza di Processo

myProcInst.Folio = "Nome comune unico del Folio"

Definizione (in C#) delle Proprietà dell'Oggetto ProcessInstance

Nome della classe (Class Name)	ProcessInstance
DataFields	public K2ROM.DataFields DataFields {get;} Esempio myValue = myProc.DataFields["FirstName"].Value.ToString();
XMLFields	public K2ROM.XmlFields XmlFields {get;} Esempio myXMLValue = myProc.XmlFields["FirstName"].Value;
Priority	public int Priority {get; set;} Esempio myProc. Priority = 1;
Folio	public string Folio {get; set;} Esempio myProc. Folio = "User Friendly Folio Name";
StartDate	public DateTime StartDate {get;} Esempio myDate = myProc.StartDate;
Name	public string Name {get;} Esempio myName = myProc. Name ;
ExpectedDuration	public int ExpectedDuration {get; set;} Esempio myDur = myProc.ExpectedDuration;
Description	public string Description {get;} Esempio myDesc = myProc.Description;
MetaData	public string MetaData {get;} Esempio myMData = myProc.MetaData;

8.4.2.10 L'oggetto Destinazione dell'Istanza di Processo (*ActivityInstanceDestination*)

L'oggetto *ActivityInstanceDestination* è accessibile chiamando la proprietà di *ActivityInstanceDestination* dell'elemento della Worklist o dell'elemento di evento Server.

Esempio :

- elemento della Worklist

myActInst = myWorkListItem.ActivityInstanceDestination

- elemento di evento server

myServerItem = myK2Conn.OpenServerItem("NumeroSeriale")

myActInst = myServerItem.ActivityInstanceDestination

Questo oggetto è usato se si vuole manipolare i campi dati dell'Attività(*Activity Data Fields*) in sostituzione ai campi dati del Processo(*Process Data Fields*). Le proprietà di questo oggetto sono accessibili nello stesso modo dell'oggetto d'Istanza di Processo, con l'eccezione che l'ID della proprietà è di sola lettura.

8.4.2.11 L'oggetto Worklist e elementi di Worklist (*Worklist and WorkListItem*)

Le proprietà, dell'oggetto di *WorkListItem* hanno alcune proprietà in più (rispetto a quelli visti fin qui), che sono di sola lettura, ma possono essere usati per trasmettere informazioni importanti. Queste proprietà includono lo **stato dell'elemento della Worklist** ma anche i dati di evento del cliente associato all'elemento della Worklist. La proprietà di dati contiene tipicamente l'URL(Universal Resource Locator) che dovrebbe essere usato per aprire il form associato all'evento dell'utente.

Si ottiene un riferimento (aggancio) alla Worklist dell'Utente di Workflow chiamando il metodo *OpenWorkList* dell'Oggetto di Connessione. Questo metodo è chiamato con un parametro di stringa indicando la piattaforma(Esempio "ASP"), e un secondo parametro facoltativo, il nome dell'utente che è manipolato dall'utente di connessione(*Connection User*).

Un determinato *WorkListItem*(elemento di worklist) può essere direttamente accessibile (senza enumerare il Worklist) chiamando il metodo **OpenWorkListItem** dell'oggetto di Connessione. Però bisogna specificare l'**unico Numero Seriale (Serial Number)** dell'elemento da aprire.

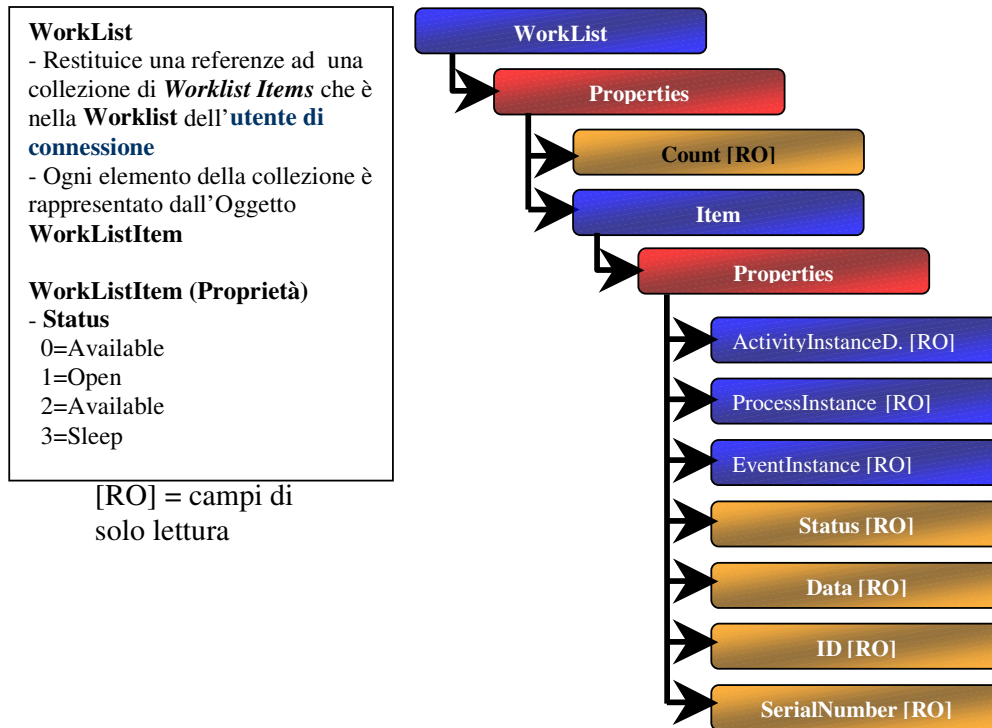


Figura 0.6:Gli Oggetti WorkList e WorkListItem

Definizione (in C#) delle Proprietà dell'oggetto **Worklist**

Nome della classe (Class Name)	Worklist
Count (numero totale di elementi della worklist)	public int Count {get;} // Sola lettura Esempio nCount = myWorkList.Count;
Item (un elemento della worklist)	public K2ROM.WorklistItem Item (int Index); {get;} // Sola lettura Esempio myWorkItem = myWorkList[n];

Definizione (in C#) delle Proprietà dell'Oggetto **WorklistItem**

Nome della classe (Class Name)	WorklistItem
ActivityInstanceDestination	public K2ROM.ActivityInstanceDestination ActivityInstanceDestination {get;} Esempio K2ROM.WorklistItem myWorkItem = myConn.OpenWorklistItem("Serial","ASP"); myActInst = myWorkItem.ActivityInstanceDestination;
ProcessInstance	public K2ROM.ProcessInstance ProcessInstance {get;} Esempio myProcInst = myWorkItem.ProcessInstance;
EventInstance	public K2ROM.EventInstance EventInstance {get;} Esempio myEventInst = myWorkItem.EventInstance;
Status	public K2ROM.WorklistStatus Status {get;} Esempio iStatus = myWorkItem.Status;
Data	public string Data {get;} Esempio sData = myWorkItem.Data;
ID	public int ID {get;} Esempio myID = myWorkItem.ID;
SerialNumber	public string SerialNumber {get;} Esempio mySerial = myWorkItem.SerialNumber;

8.4.2.12 Metodi della “WorklistItem “

I metodi di WorklistItem sono usati per progettare le simulazioni delle azioni intraprese in background (dietro le quinte) quando un'utente di workflow accede alla sua Worklist con il K2.net Workspace e apre/complete alcune elemento della Worklist.

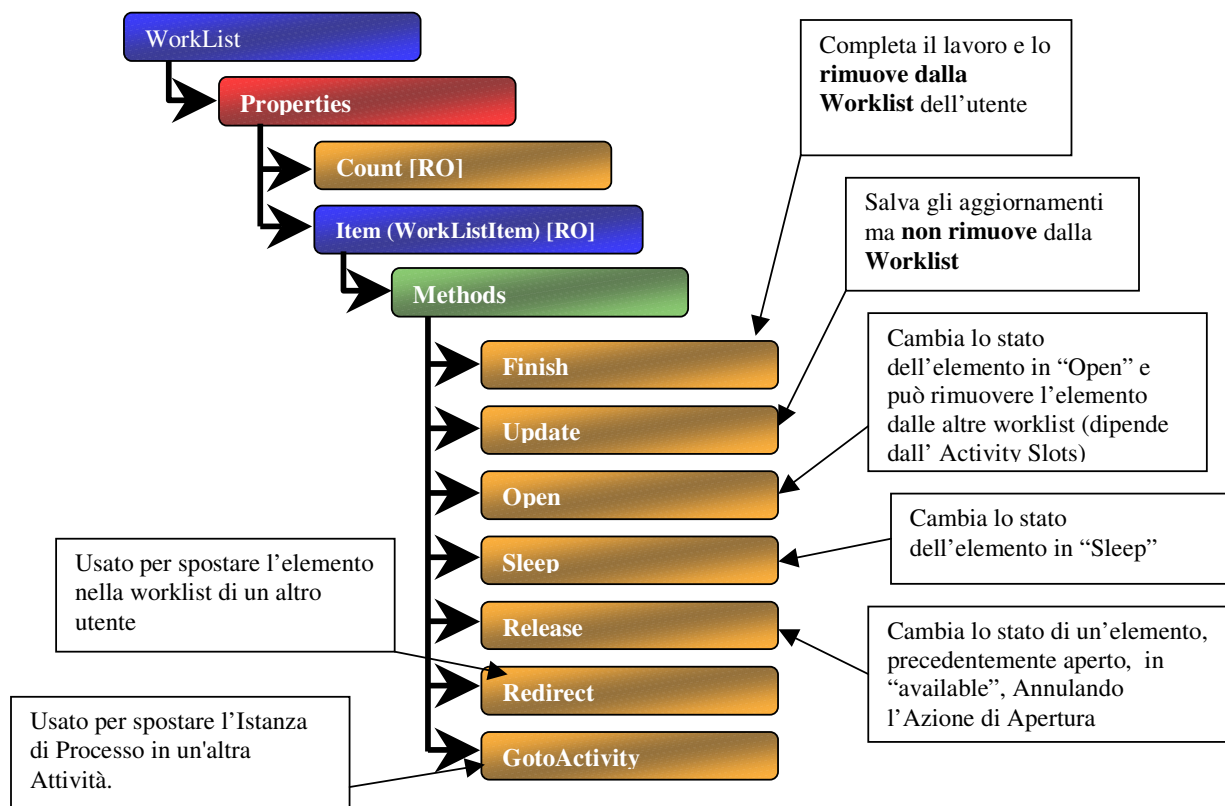


Figura 0.7: Metodi della "WorkListItem"

Definizione (in C#) delle Proprietà del metodo "WorkListItem"

Nome della classe (Class Name)	WorkListItem
Finish	public void Finish (); public void Finish (bool Sync); Esempio myWorkItem.Finish();
Update	public void Update (); Esempio myWorkItem.Update();
Open	public void Open (); Esempio myWorkItem.Open();
Sleep	public void Sleep (bool Flags);

	<pre>public void Sleep (bool Flags,int Sleep);</pre> <p>Esempio</p> <pre>myWorkItem.Sleep(true,86400);</pre>
Redirect	<pre>public void Redirect (string User);</pre> <p>Esempio</p> <pre>myWorkItem.Redirect("KellyJ");</pre>
GotoActivity	<pre>public void GotoActivity (string Activity);</pre> <pre>public void GotoActivity (string Activity ,bool Sync);</pre> <p>Esempio</p> <pre>myWorkItem.GotoActivity("Manager Approval");</pre>

Riassunto dei punti importanti

Come visto in precedenza K2ROM risulta essere lo strumento indispensabile per fare interagire un software esterno con K2.Net . Nel prossimo capitolo sarà studiato il modo in cui questo avverrà partendo da un esempio.

9. ANALISI INTEGRAZIONE DI K2.NET CON SOFTWARE ESTERNI

9.1 Introduzione

In questa fase della tesi viene studiato il modo in cui un software esterno, sviluppato da un'azienda, può interagire (integrarsi) con i processi Workflow di K2.net 2003.

I punti in cui il software esterno deve interagire con K2.net 2003 sono distribuiti in tre moduli:

- Input Form Generator (IFG): software per la creazione automatica del form web creando una tabella con i vari campi.
- Autorizzazioni: software di autorizzazione ai vari servizi e contenuti internet
- Creamoduli: software di gestione automatica di tutti i campi delle tabelle di un determinato database

9.2 Analisi dell'integrazione

Prima di procedere bisogna illustrare le richieste dell'azienda che, in questo studio ha sviluppato dei servizi (CMS, Sportello sociale, URP...) accessibili da web tramite portali internet e reti civiche. Le tre applicazioni citate sopra (IFG, Autorizzazione, Creamodulo), già funzionanti saranno usate per interagire con il software di K2.net 2003. il primo per creare le tabelle con tutti i dati necessari, il secondo per gestire le autorizzazioni agli utenti del Portale internet e il terzo per la gestione dei campi delle tabelle contenenti le informazioni sui processi Workflow.

9.2.1 Installazione dei componenti di K2.net 2003:

Prima di tutto l'azienda vuole gestire tutti i processi Workflow dei clienti dai suoi server. Di conseguenza i componenti server (tra cui K2.net 2003 server...) del software di K2.net 2003 saranno installati sui server (accessibili dal web) dell'azienda che gestirà e renderà accessibili i processi Workflow, invece i componenti client (tra cui K2.net 2003 studio...)

del software di K2.net 2003 saranno installati sui pc (server) dei clienti che dovranno esportare i loro progetti Workflow sui server dell'azienda dall'area riservata di un sito internet/intranet. Lo scenario è illustra nella seguente figura :

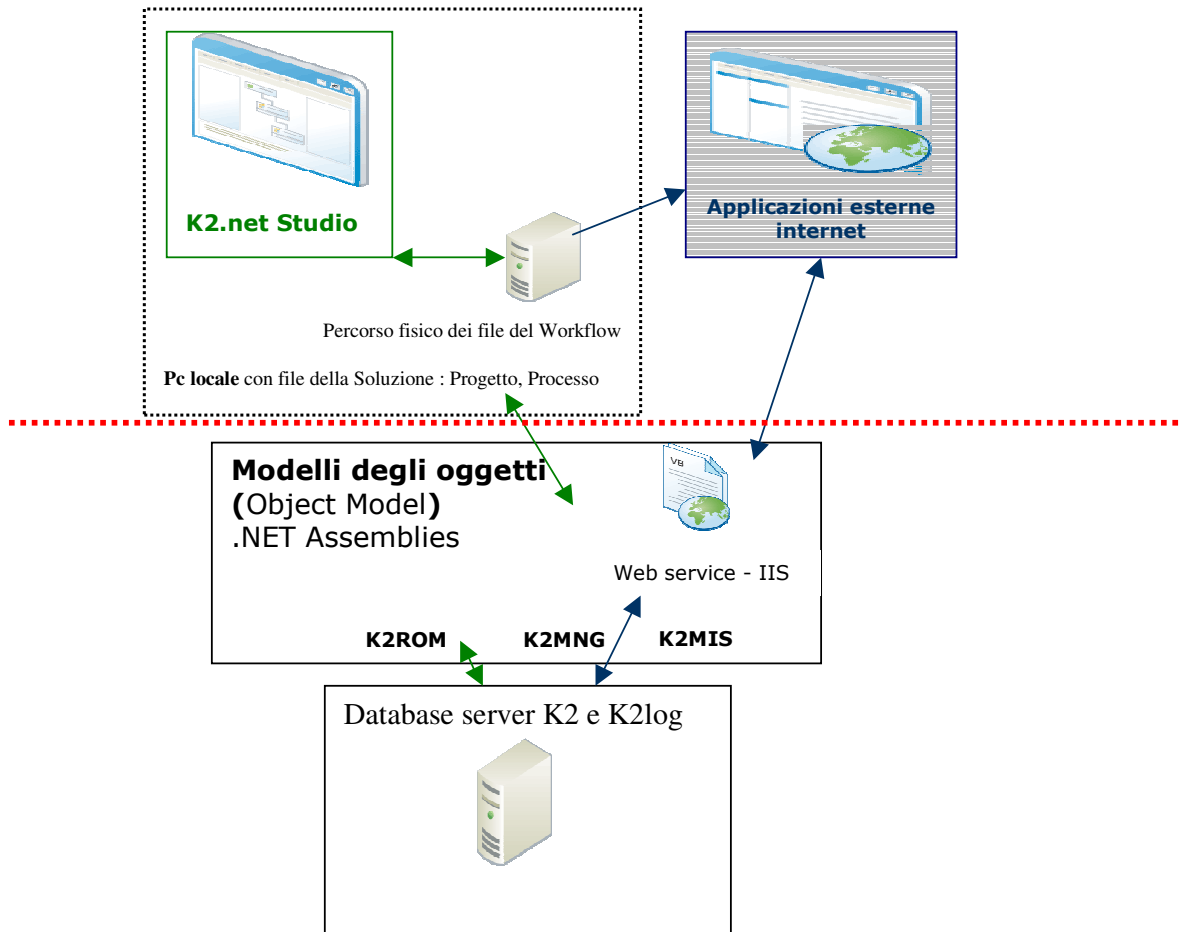


Figura 9.1: interazione tra applicazione esterno e K2.net per la gestione di progetti Workflow

9.2.2 Gestione degli utenti

Sui server dell'azienda non è installato l'Active Directory. Di conseguenza è necessario installare K2.net 2003 SQL User Manager che consente di gestire gli utenti e gruppi di utenti da un database separato(SQLUM). Dall'analisi fatta sul SqlUserManager è risultato che la copertura è totale, è possibile quindi fare a meno dell'Active Directory di Windows ed utilizzare delle tabelle di SqlServer per la gestione degli utenti in K2. Tutti i componenti di K2 (Studio, Workspace e Service Manager) leggeranno quindi gli utenti e i gruppi dalle tabelle di SqlServer, presenti nel Database di K2(SQLUM), e non da Active Directory.

9.2.2.1 Integrazione tra database utenti del portale e quello di K2.net

Dopo aver analizzato la flessibilità di configurazione dei software di K2, è risultato che è possibile, per K2.net 2003 Studio e K2.net 2003 Workspace, configurare il Database degli utenti a cui essi si devono collegare per leggere appunto l'elenco degli utenti e dei gruppi. Per integrare utenti e gruppi di K2 con quelli già presenti in database diversi, si è quindi deciso di :

1. mantenere nel Database principale(SQLUM) di K2 l'elenco di tutti gli utenti dei Portali,
2. creare una copia del Database di SQLUM per ogni Portale internet. I dati degli utenti di ogni portale sono memorizzati nel Database chiamato per esempio DatabasePortale1_SQLUM.

Essendo necessario uno username univoco tra gli utenti e i gruppi di K2 si potrebbe abbinare all'username dell'utente il nome del database del portale internet come illustrato: <username>@< DatabasePortale1>. Dove username identifica in modo univoco ogni utente del Portale nel database DatabasePortale1.

I dettagli di questa scelta saranno illustrati nel prossimo capitolo con lo studio del caso SUAP.

9.2.2.2 Autorizzazioni a utenti o gruppi

K2 offre una ricca gestione delle autorizzazioni degli utenti ad intervenire nelle fasi di un processo. Gli utenti possono appartenere a gruppi. Ogni utente ha una worklist che contiene tutte le attività di workflow che deve effettuare. Questa gestione è insostituibile, e deve guidare e controllare tutto quello che avviene agli elementi sottoposti a workflow. Il controllo delle autorizzazioni finali (comprese quelle di K2) di un utente è a carico del servizio *Autorizzazioni*. Questo servizio già esiste per la gestione delle autorizzazioni nei vari portali internet sarà usato anche per gestire gli accessi degli utenti del workflow dal portale. Per esempio un utente registrato nel portale internet può essere autorizzato ad accedere come attore di un Processo workflow direttamente da internet. È chiaro che i dati (username, password, e-mail ...) di tale utente saranno esportati nel database degli utenti di K2.net

9.3 Integrazione Dati

Le applicazioni dell'azienda gestiscono dei dati memorizzati in vari database, dall'altra parte K2 ha un suo database che permette di memorizzare dei dati associati ad ogni istanza attiva del processo di workflow, ma non è pensabile passare a K2 tutti i dati gestiti dalle applicazioni dell'azienda.

9.3.3 Requisiti per le applicazioni esterne a K2.net 2003

Le applicazioni dell'azienda devono "agganciarsi" alle istanze di workflow, oppure devono avviare dei nuovi processi (per esempio da un utente del portale dopo aver compilato un apposito form). A tal fine è necessario memorizzare dentro il database dell'azienda gli identificatori degli elementi di K2 a cui ogni record del database è correlato. Quando un dato record è collegato ad una istanza del processo, l'utente deve anche avere la possibilità di aggiornarlo senza per forza fare avanzare il flusso (ad esempio se ci sono dati incompleti). Questa funzionalità va apportata in tutti i servizi che interagiscono col workflow: IFG(input form generator che crea i form con le relative tabelle e campi), creamoduli (che gestisce le tabelle e i loro campi).

L'istanza di workflow deve ricevere dall'applicazione dell'azienda tutti i dati che sono necessari e sufficienti per il proseguimento del processo, ad esempio se la determinazione della fase da eseguire dipende dal valore immesso in un campo dall'utente finale, questo

valore deve essere passato a K2 – ed ovviamente deve essere previsto nella fase di design del processo.

Gli elementi da sviluppare sono quindi quattro:

1. quali campi del database dell'azienda devono contenere i riferimenti al processo di workflow
2. modalità di aggiornamento (deploy) dei campi(del database dell'azienda) precedenti
3. quali campi del processo di workflow contengono i dati immessi dall'utente dell'applicazione esterna dell'azienda (almeno quelli necessari al processo)
4. modalità di aggiornamento (deploy) dei campi(del processo di workflow) precedenti

9.3.4 Interfaccia da applicazioni esterne a K2.net 2003

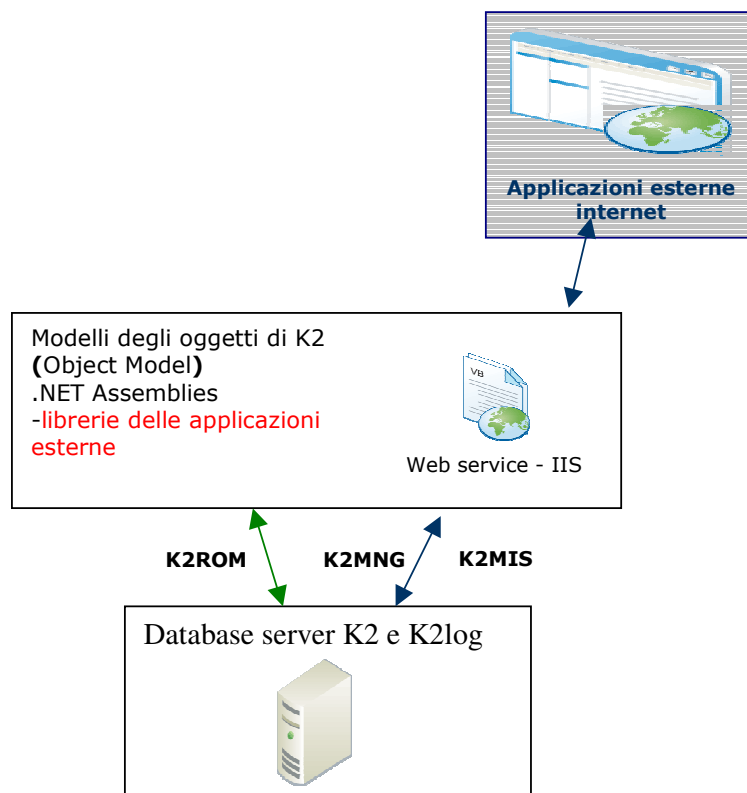


Figura 9.2 :Librerie di interfacciamento tra le applicazioni esterne e di K2.net

L'azienda deve sviluppare delle librerie di interfacciamento tra K2 e le sue applicazioni e viceversa.

L'azienda deve essere in grado di :

- ❑ accedere al server K2.net 2003
- ❑ vedere tutti i processi disponibili per un determinato cliente
- ❑ accedere alle proprietà del processo (essere in grado di agganciare i campi del software aziendale a quelli del flusso): campi, attività, eventi
- ❑ avviare una nuova istanza di un certo processo passandogli gli opportuni dati
- ❑ agganciarsi ad una istanza di processo (in realtà all'evento che deve essere eseguito in quel momento), leggerne i dati e farla avanzare, passandogli gli opportuni dati
- ❑ accedere alla worklist di un dato utente
- ❑ controllare i permessi di un utente definiti dal service manager (da verificare)
- ❑ controllare la worklist di un utente

9.4 Analisi struttura database K2.net

9.4.1 Database Utenti

Visto che durante lo studio dell'implementazione di K2.net 2003 abbiamo preso in considerazione un'ambiente software privo di Active Directory. Di conseguenza abbiamo installato il componente K2.net SQLUM che crea automaticamente un database K2SQLUM come visto in precedenza. La struttura di tale database è descritta di seguito

Struttura del database (K2SQLUM) dove sono memorizzati gli utenti che possono accedere a K2.net 2003. Il database è composto da tre tabelle **K2Users**, **K2UserGroup** e **K2Groups**

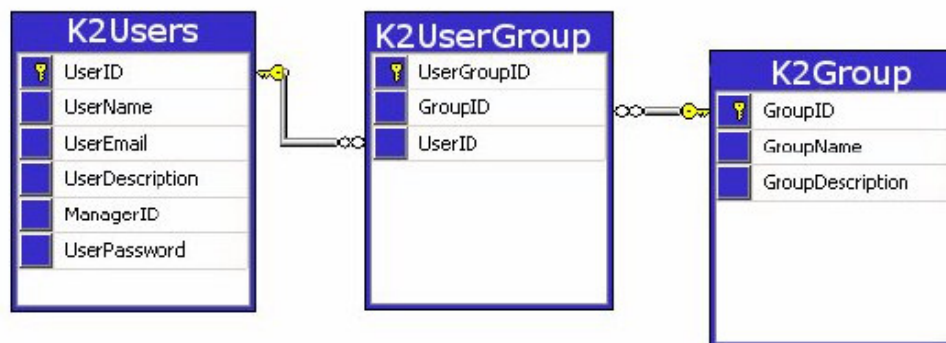


Figura 9.3: diagramma del database utenti K2SQLUM

```
CREATE TABLE [dbo].[K2Groups] (
  [GroupID] [int] IDENTITY (1, 1) NOT NULL ,
  [GroupName] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
  [GroupDescription] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[K2UserGroup] (
  [UserGroupID] [int] IDENTITY (1, 1) NOT NULL ,
  [GroupID] [int] NULL ,
  [UserID] [int] NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[K2Users] (
  [UserID] [int] IDENTITY (1, 1) NOT NULL ,
  [UserName] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
  [UserEmail] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [UserDescription] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
  [ManagerID] [int] NULL ,
  [UserPassword] [varchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

La tabella K2Users contiene i dati degli utenti che appartiene ad un gruppo di utenti descritto tramite

The screenshot displays three data windows from SQL Server Enterprise Manager:

- Data in Table 'K2Users' in 'K2SQLUM' on '(local)'**: Shows two rows of user data.

UserID	UserName	UserEmail	UserDescription	ManagerID	UserPassword
1	Administrator	Administrator@server.it	Utente amministratore	<NULL>	administrator
2	bertrand@um_2	bertrandt@yahoo.it	amministratore	<NULL>	bertrand
- Data in Table 'K2UserGroup' in 'K2SQLUM' on '(local)'**: Shows one row linking a user to a group.

UserGroupID	GroupID	UserID
1	1	2
- Data in Table 'K2Groups' in 'K2SQLUM' on '(local)'**: Shows one row for a group.

GroupID	GroupName	GroupDescription
1	PDIGroup	Gruppo PDI

Figura 9.4: Esempio contenuto tabelle utenti

In questo esempio l'utente [bertrand@_um2](#) con UserID uguale a 2 fa parte del gruppo PDIGroup con GroupID uguale a 1

9.4.2 Database delle transazioni chiamato K2

In questi database (in totale due chiamati K2 e K2log) creati automaticamente durante l'installazione del componente K2.net 2003 server viene memorizzata tutta la struttura di una soluzione Workflow (Dai processi agli eventi passando dalle varie regole e proprietà delle varie linee di collegamento alle Attività) e tutte le transazioni operate. Di seguito verrà illustrata la struttura delle tabelle relative ai processi, Attività e eventi. Da notare che tutte le tabelle del database iniziano con il simbolo "underscore". Esempio la tabella delle Attività "act". Il primo database K2 serve per memorizzare la struttura completa dei processi dalle attività alle variabili usati. Il secondo database K2log con la struttura identica contiene in più i dati relativi alle istanze di processo.

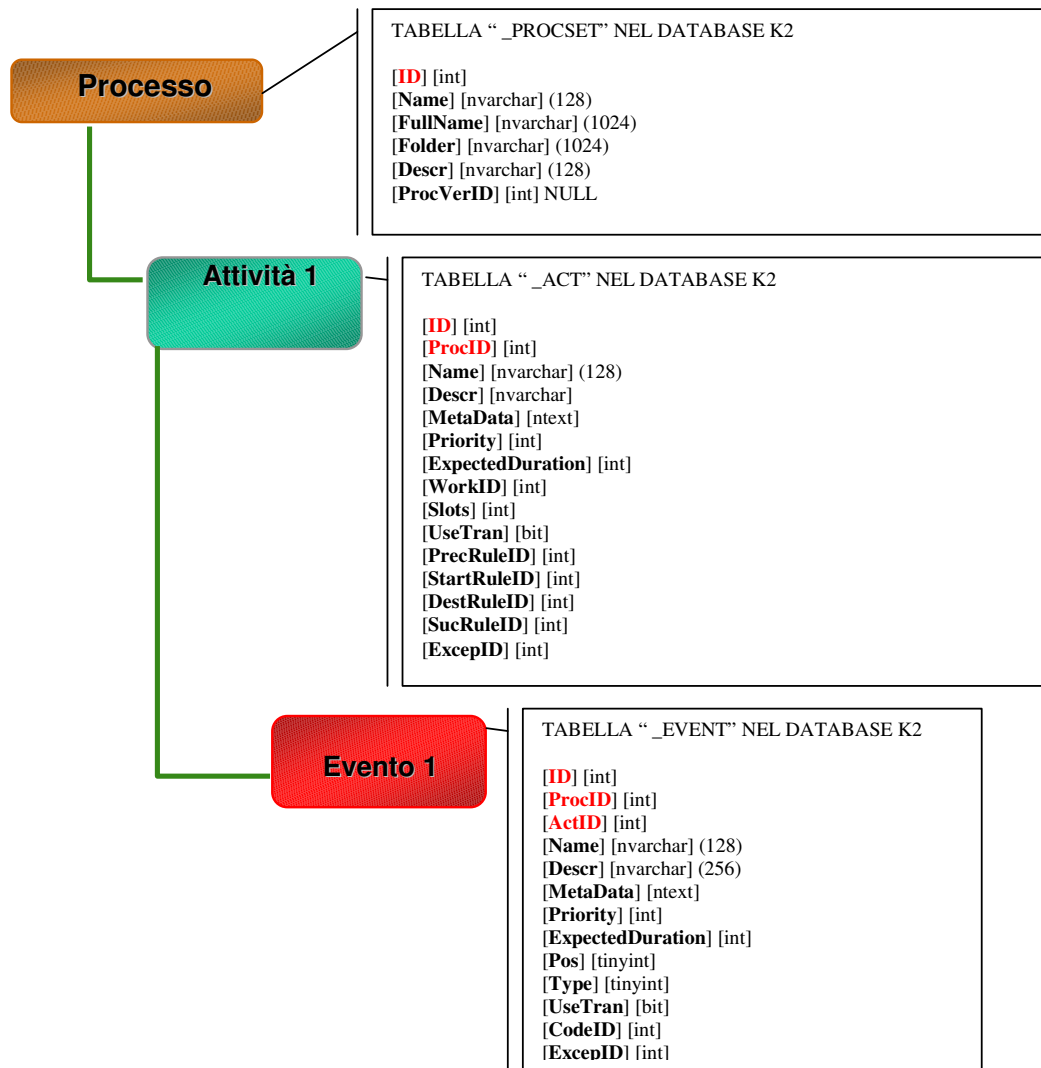


Figura 9.5: Identificazione di Processi, Attività e Eventi nel Database K2

9.4.2.1 Tabelle dei processi

Come spiegato nei capitoli precedenti i processi hanno varie versioni, di conseguenza nel database ci sono due tabelle (`_Procset` e `_Proc`) che consentono di memorizzare informazioni sui processi (versione, data esportazione ecc). la prima tabella, `_Procset` contiene i dati originali del processo come nome, locazione e soprattutto la versione attuale

inserito nel campo “**ProcVerID**” che è sempre l’ultima versione del processo esportato da K2.net 2003 Studio. Vediamo un esempio di contenuti di queste due tabelle:

Tabella “_Procset”

ID	Name	FullName	Folder	Descr	ProcVerID
1	sale	Sales\sale	Sales		2

Da notare che il nome completo del Processo è composto dal nome del progetto (che è anche la cartella dove è salvato il processo) in questo esempio “Sales” il nome del processo stesso. La versione attuale del processo è 2 il che significa che c’è una versione 1.

Alcuni campi della Tabella “_Proc”

ID	ProcSetID	ExportID	Priority	ExpectedDuration	WorkID	LogLevel	Language	Ver	ChangeDate
1	1	1	0	0	0	3	1	1	24/07/2006
2	1	2	0	0	0	3	1	2	24/07/2006

Come si vede dalla tabella di sopra il Processo con ProcSetID =1 ha due versioni.

9.4.2.2 Tabella dell’Attività

Un’attività è sempre collegato ad un Processo tramite la chiave esterna “*ProcID*” che fa riferimento al ID della tabella “_Proc” delle versioni dei processi

Alcuni campi della Tabella “_Act”

ID	ProcID	Name	WorkID	Slots	UseTran	PrecRuleID	StartRuleID	DestRuleID	SucRuleID
1	1	Booking Made	0	1	FALSO	0	0	0	0
2	1	Manager Approval	0	1	FALSO	2	0	3	4
3	1	Make Booking	0	1	FALSO	6	0	7	8
4	1	VP approval	0	1	FALSO	10	0	11	12
5	1	Start	0	1	FALSO	0	0	0	0
6	1	Declined	0	1	FALSO	0	0	0	0
7	2	Make Booking	0	1	FALSO	21	0	22	23
8	2	Booking Made	0	1	FALSO	0	0	0	0
9	2	Manager Approval	0	1	FALSO	26	0	27	28
10	2	VP approval	0	1	FALSO	30	0	31	32
11	2	Start	0	1	FALSO	0	0	0	0
12	2	Declined	0	1	FALSO	0	0	0	0

Come si vede dalla tabella ” _act” i record di tutte Attività sono sempre salvate di nuovo tutte le volte che viene creata (esportata) una nuova versione del processo, aggiornando i vari campi. Per esempio le Attività con ID 6 e 12 sono le stesse ma fanno riferimento a due versioni dello stesso processo (con nome “Sale” nella tabella “_Procset”) con ProID uguale a 1 e 2 rispettivamente.

9.4.2.3 Tabella degli Eventi

Un’Evento, come un’Attività per un processo, è sempre collegato ad un’Attività tramite la chiave esterna “ActID” che fa riferimento al ID della tabella “_Act”.

Alcuni campi della Tabella “_Event”

ID	ProcID	ActID	Name	Priority	ExpectedDuration	Pos	Type	UseTran	CodeID
1	1	1	Booking confirmation	0	0	1	1	FALSO	1
2	1	2	Manager decision	0	0	1	2	FALSO	5
3	1	3	Booking details	0	0	1	2	FALSO	9
4	1	4	VP decision	0	0	1	2	FALSO	13
5	1	6	Reject notification	0	0	1	1	FALSO	14
6	2	7	Booking details	0	0	1	2	FALSO	24
7	2	8	Booking confirmation	0	0	1	1	FALSO	25
8	2	9	Manager decision	0	0	1	2	FALSO	29
9	2	10	VP decision	0	0	1	2	FALSO	33
10	2	12	Reject notification	0	0	1	1	FALSO	34

Come si è visto per le attività ogni evento viene salvato di nuovo tutte le volte che viene creata (esportata) una nuova versione del processo. Per esempio l’Evento con ID 4 e 9 sono gli stessi ma fanno riferimento a due versioni dello stesso processo (con nome “Sale” nella tabella “_Procset”) con ProID uguale a 1 e 2 rispettivamente.

10. ANALISI E STRUTTURA DEI DATABASE DELLE APPLICAZIONI ESTERNE A K2.NET 2003

10.1 Introduzione

Il servizio (applicazione) WORKFLOW è una funzionalità da sviluppare dall'azienda che vuole fare interagire i record (dati) delle sue applicazioni con i processi Workflow definiti con K2.net 2003. Per motivi di sicurezza quest'applicazione sarà accessibile solo dall'area riservata (Backoffice) di un sito internet o intranet.

Questa funzionalità ha lo scopo di integrare un servizio (esempio : notizie, CRM, Gestione documentale...) con la soluzione K2 per la gestione dei Workflow.

Tramite questo servizio si dovranno quindi mappare su Database aziendale i seguenti legami con K2:

1. il legame **tabella** dell'applicazione - **Processi K2** (0 a n): una qualsiasi tabella dati deve poter essere associata a nessuno o ad n processi definiti con K2;
2. il legame **record** di una tabella dell'applicazione - **Processi K2** (0 a n): i record di una tabella potranno quindi appartenere a più processi contemporaneamente;
3. il legame **campo** di una tabella dell'applicazione - **variabili di processo K2** (0 a n): l'associazione tra i campi di una tabella e le variabili di processo, per ogni processo a cui la tabella è associata;
4. il legame **campo** di una tabella dell'applicazione - **attività di Processo K2** (0 a n): per ogni campo sarà possibile indicare se è visibile e/o modificabile ad ogni attività di ogni processo.

NOTA:

In questa tesi vengono analizzati solamente i tracciati record necessari per mappare questa integrazione; non viene quindi trattata l'analisi completa del software da sviluppare.

10.2 Implementazione della struttura una *Solution* (Soluzione) workflow di K2.net nel database esterno

Si decide di mantenere la definizione della *Solution* di K2 in tabelle apposite. Successivamente i legami dei servizi con il workflow avranno riferimenti in queste tabelle.

Il file della *Solution* viene inviato al server (come illustrato nella figura precedente) che ospita il portale tramite un servizio opportuno, che si occupa di accedere al file e di leggerne la definizione: progetti, processi, attività e variabili. Lo stesso servizio si occupa del deploy (trasferimento) sul server K2 dei processi contenuti nel file di solution inviato (con modalità da definire, e comunque al di fuori dallo scopo di questa Tesi). In questo modo si ha il grosso vantaggi che non viene fornito all'utente un accesso diretto al server K2 per il deploy dei processi questo per aumentare la sicurezza e rendere il trasferimento più semplice .

Per convenzione tutte le tabelle interessate dal presente capitolo iniziano per TWF_K2. Inoltre si rispettano tutte le convenzioni in essere nel servizio “*creamodulo*”, specificatamente per mappare ciascuna tabella in un menù a tendina e sulla determinazione delle FK.

Nota: il servizio *creamodulo* di proprietà di Progetti di impresa srl è un programma che ha il compito di gestire in modo automatico tutti i campi, con le loro proprietà (tipo del campo, dipendenza ad un'altra tabella, ecc...), delle tabelle di un database. Questo consente di mappare tutte le caratteristiche delle tabelle (Esempio: numero di campi di una tabella, posizione del campo) e delle tabelle associate ad un determinato campo create automaticamente.

Segue un paragrafo per ciascuna tabella, in ordine gerarchico.

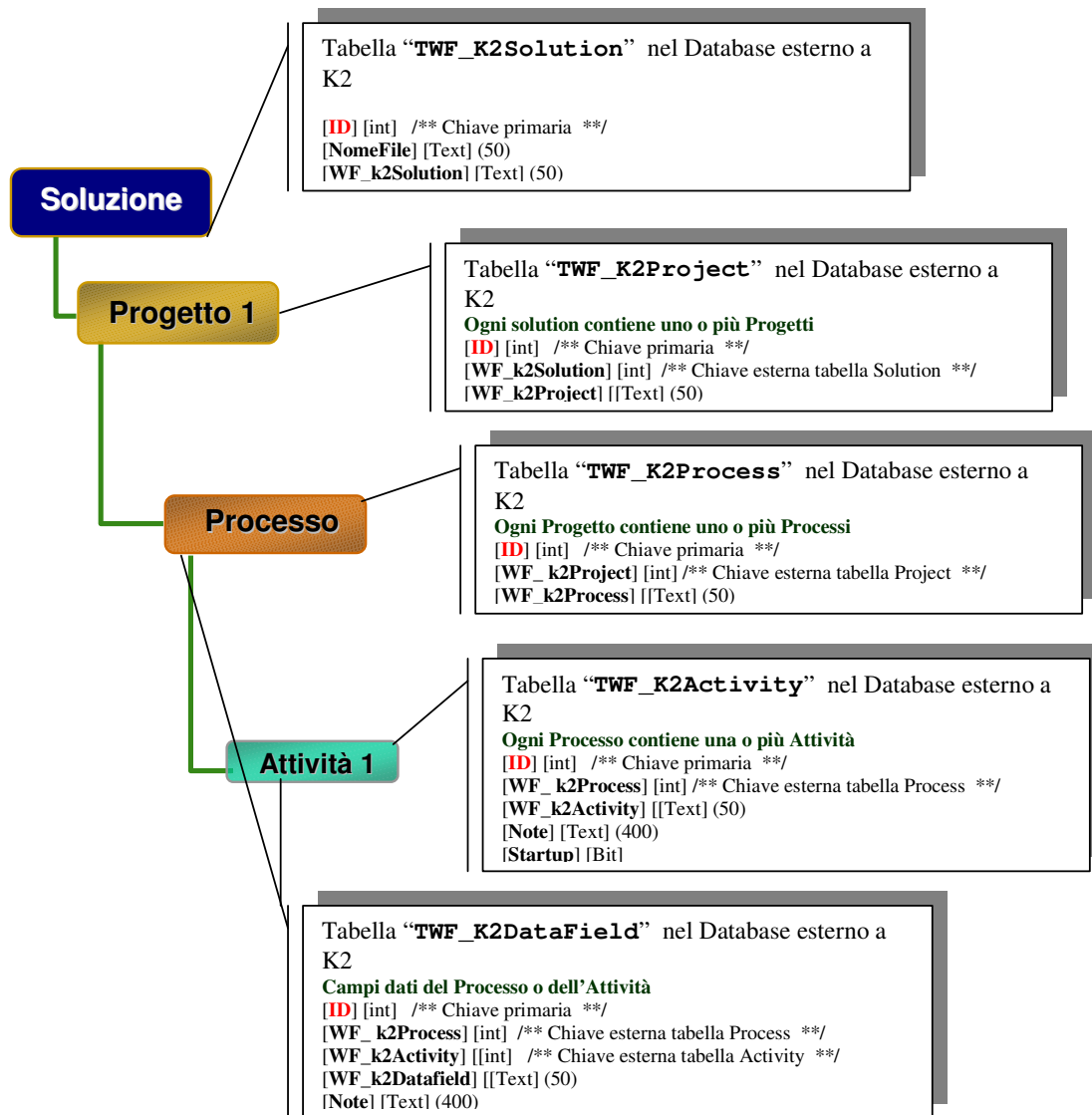


Figura 10.1: Soluzione Workflow vista da un Database esterno a K2.net 2003

10.2.1 Tabella della soluzione *TWF_K2Solution*

Come evidenziato nelle figure ogni *Solution* disegnato da K2.net Studio contiene uno o più progetti separati. Però nel database di K2.net non c'è una tabella riservata alle Solution in quanto viene identificata da un file fisico e gestita a livello software. Questa tabella serve all'applicazione esterna per identificare i vari progetti della soluzione.

10.2.2 Tabella del progetto *TWF_Project*

Ogni progetto contiene uno o più processi. Questa tabella serve all'applicazione esterna per identificare i vari processi del progetti.

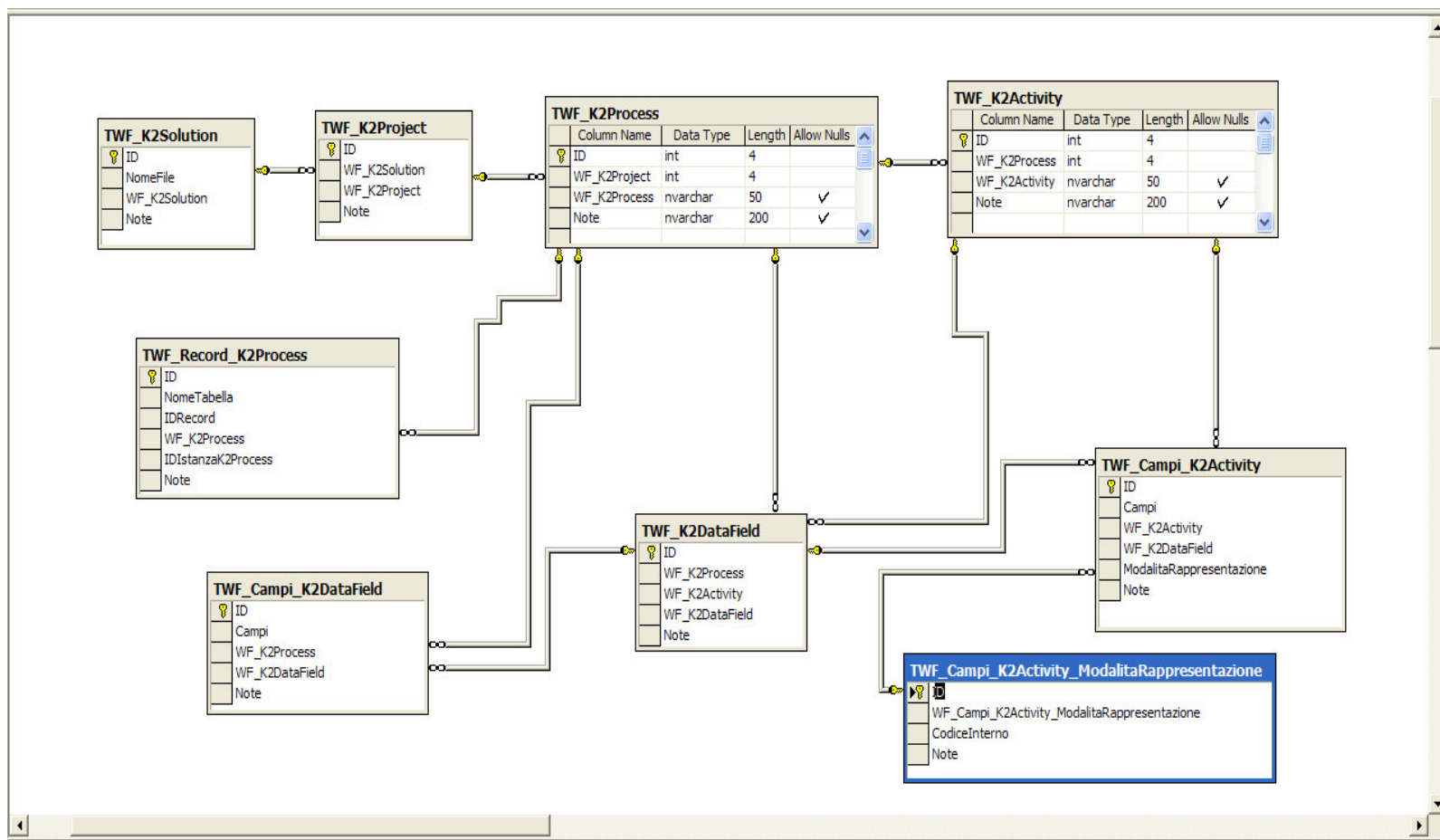


Figura 10.2: Diagramma del Workflow visto da un database esterno a quello di K2.net

10.2.3 Tabella dei processi *TWF_K2Process*

Da K2.net ogni processo viene identificato dal Fulname “**nomeprogetto\nomeprocesso**” come descritto nella tabella “_Procset” del database K2

WF_K2Project	int	FK su TWF_Project.ID
WF_K2Process	Testo breve	ID del processo (interno di K2) creato con il Designer di K2, nel formato nomeprogetto\nomeprocesso .

Struttura tabella Processi

10.2.4 Tabella delle attività *TWF_K2Activity*

Ogni processo contiene una o più attività.

WF_K2Process	int	FK su TWF_Project.ID
WF_K2Activity	Testo breve	Nome dell'attività come definito in K2.net Studio
Startup	bit	Serve per identificare l'attività iniziale di un processo

Struttura tabella delle attività

10.2.5 Tabella dei campi del processo/attività

TWF_K2DataField

Tabella dei Campi che memorizzano i dati di processo o attività “TWF_K2datafield” .

In questa tabella devono essere inseriti tutti le variabili del processo definiti in K2.net

Column Name	Data Type	Length	Allow Nulls
ID	int	4	
WF_K2Process	int	4	✓
WF_K2Activity	int	4	✓
WF_K2DataField	nvarchar	50	✓
Note	nvarchar	200	✓

Figura 10.3: Tabella TWF_K2DataField

WF_K2Process	intero	FK su TWF_K2Process.ID. Può essere NULL, in questo caso l'altro campo sarà NOT NULL e indica una variabile dell'attività indicata da K2Activity.
WF_K2Activity	intero	FK su TWF_K2Activity.ID. Può essere NULL, in

		questo caso l'altro campo sarà NOT NULL e indica una variabile del processo indicato da K2Process.
WF_K2DataField	Testo Breve	Nome del campo indicato da K2 Studio.

Struttura tabella dei campi del processo/attività

10.2.6 Tracciati record

Di seguito sono riportate le tabelle di sistema da aggiungere a quelle già esistenti per poter mappare i legami indicati. Nei tracciati record non sono indicati I vari campi di sistema comuni tra tutte le tabelle (ID, datainserimento, ecc...).

Per convenzione tutte le tabelle interessate dal presente capitolo iniziano per TWF_.

10.2.6.1 Legame tabella Creamodulo – Processi K2 (0 a n)

Una tabella può essere associata a 0 o n processi di K2, la seguente tabella mappa quindi il nome della tabella e il codice identificativo del processo. Questi sono i processi K2 abilitabili su una singola tabella creata con il servizio IFG(Input form generator) che ha il compito di creare una tabella con vari campi impostati dall'utente tramite un form.

TWF_Tabelle_K2Process

NomeTabella*	Testo breve	Nome della tabella gestito con il Creamodulo.
WF_K2Process*	Intero	FK su TWF_K2Process.ID.

Le colonne indicate con (*) sono Alternate Key della tabella.

Esempio:

Tabella TWF_Tabelle_K2Process

ID	WF_K2Process	NomeTabella	Note
1	1	NomeTabellaCreato	

Come si vede in questo esempio la tabella “NomeTabellaCreato”, creata dal servizio IFG tramite, è associata al processo con ID=1 nella tabella dei processi T_WF_K2Process.

10.2.6.2 Legame record di una tabella – Processi K2 (0 a n)

Il database viene profilato in modo che non escluda la possibilità di avere un record di IFG associato a due differenti istanze di processo K2.

I processi a cui i record di una tabella possono essere soggetti sono quelli definiti nella tabella *TWF_tabelle_K2process*. E la tabella dei record in questione è chiamata *TWF_Record_K2Process* come descritta di seguito.

TWF_Record_K2Process

NomeTabella*	Testo breve	Nome della tabella gestita con il Creamodulo
IDRecord*	Intero	ID del record della tabella
WF_K2Process*	Numerico	FK su <i>TWF_K2Process.ID</i> .
IDIstanzaK2Process*	Numerico	ID dell'istanza di processo assegnato da K2

Le colonne indicate da (*) sono Alternate Key della tabella.

Il legame avviene tramite i campi:

- NomeTabella, IDRecord è L'ID del record nella tabella,
- IDProcesso, IDIstanzaProcesso sono assegnati da K2

Esempio:

Contenuto tabella *TWF_Record_K2Process*

ID	NomeTabella	IDRecord	WF_K2Process	IDIstanzaK2Process	Note
1	NomeTabellaCreata	3	1	27	
2	NomeTabellaCreata	4	1	31	
3	NomeTabellaCreata	5	1	32	
4	NomeTabellaCreata	6	1	33	

In questo esempio Ci sono quattro record della tabella *NomeTabellaCreata* che sono assegnati al Processo con ID=1 nella tabella *TWF_K2Process*. Ad ogni di questi record è stato assegnato (Da K2.net) un unico ID di istanza del processo *IDIstanzaK2Process*

10.2.6.3 Legame tra campo di una tabella e variabili di Processo/Attività K2 (0 a n)

Una volta definiti i processi a cui una tabella può essere soggetta, per ognuno di essi deve essere mappato quale campo della tabella è legato a quale variabile di ogni processo. Si

decide che la mappatura *campo tabella creata con IFG e variabile di processo* non dipende dalla activity in cui si trova il processo.

NB: le variabili di processo/attività devono essere definite prima in *K2.net Studio*, poi nell'IFG tramite un wizard che scrive la mappatura del campo nella seguente tabella. Se viene aggiunta, eliminata o modificata successivamente una variabile di processo in *K2.net Studio*, è necessario riconfigurare nell'IFG, sempre tramite wizard, il legame tra campi e variabili di processo.

Il legame viene effettuato con la variabile indicata da TWF_K2DataField.ID, che può essere una variabile definita a livello di process o activity nel K2 designer.

Definizione della tabella TWF_Campi_K2DataField

Nome Campo	Tipo campo	Descrizione del campo
Campi	Intero	FK = ID del campo nella tabella creamodulo.
WF_K2Process	Intero	FK su TWF_K2Process.ID.
WF_K2DataField	Intero	FK su TWF_K2DataField.ID.

Il campo *Campi* identificati in modo univoco(ID) un campo (con tutte le sue proprietà: tabella associata, tipo...) nella tabella *Creamodulo*

Esempio :

contenuto tabella *TWF_Campi_K2DataField*

ID	Campi	WF_K2Process	WF_K2DataField	Note
1	23	1	3	

Il Datafield con ID =1 è associato al Processo con ID=1.

Per sapere se tale campo è associato direttamente al processo o attività basta ricerca la riga corrispondente al WF_K2DataField = TWF_K2DataField .ID=3 .

Dalla tabella di sotto risulta che il campo appartiene al processo e si chiama *Cognome*.

Contenuto tabella TWF_K2DataField

ID	WF_K2Process	WF_K2Activity	WF_K2DataField	Note
3	1		Cognome	

Per sapere tutte proprietà(Tipo, data creazione...) del campo *Cognome* basta vedere il record con ID= 23 nella tabella Creamodulo.

Esempio Contenuto tabella Creamodulo

ID	NomeTabella	NomeCampo	TipoCampo	Note
23	NomeTabellaCreato	Cognome	Int	

10.2.6.4 Visibilità tra campo di una tabella Creamodulo e attività di Processo K2

Per l'integrità e la riservatezza di alcuni campi si può decidere di nascondere o meno tali campi al momento del collegamento con le attività. Per esempio un campo può essere modificabile in una determinata attività ma essere di solo lettura o nascosta in un'altra attività.

Per ogni tabella IFG, sarà possibile impostare, per ogni Attività di ogni processo a cui è stata associata tramite *TWF_Tabelle_K2Process*, la modalità di rappresentazione di ogni singolo campo. Le possibili modalità di rappresentazione sono:

- Campo nascosto (Di conseguenza non sarà visibile nell'Attività)
- Campo visibile, ma non modificabile (sola lettura)
- Campo visibile e modificabile

Se un campo non viene mappato in questa tabella(*TWF_Campi_K2Activity*), per default si assume che in tutte le attività tale campo sia visibile e modificabile.

Definizione della Tabella *TWF_Campi_K2Activity*

Nome Campo	Tipo Campo	Descrizione Campo
Campi	Intero	FK = ID del record della tabella Creamodulo
WF_K2Activity	Intero	FK su <i>TWF_K2Activity.ID</i> . Attività su cui si deve applicare una modalità di rappresentazione particolare per il campo indicato.
ModalitaRappresentazione	Intero (Menù a tendina)	Indica la modalità di rappresentazione del record quando sarà giunto all'attività indicata
WF_K2DataField	Intero	FK su <i>TWF_K2DataField.ID</i> . indicata.

a) Tabella della modalità di rappresentazione dei campi

TWF_Campi_K2Activity_ModalitaRappresentazione è la Tabella dei contenuti del menù a tendina che imposta il valore nel campo *TWF_campi_K2Activity.ModalitaRappresentazione*, contenente le voci:

- Campo nascosto
- Campo visibile, ma non modificabile (sola lettura)
- Campo visibile e modificabile

Il software è strettamente legato a queste voci, infatti il suo comportamento è pilotato dal valore presente in questa tabella per un dato campo. Il collegamento tra il software di gestione del Workflow ed i dati su database è dato dal campo *CodiceInterno*.

La struttura della tabella è quindi:

Definizione della Tabella *TWF_campi_K2Activity_ModalitaRappresentazione*

Nome Campo	Tipo Campo	Descrizione Campo
<i>WF_Campi_K2Activity_ModalitaRappresentazione</i>	testo breve	Voce visualizzata nel menù a tendina
<i>CodiceInterno</i>	testo breve (10 caratteri)	Codice interno associato alla voce, pilota il funzionamento del software
<i>Note</i>	testo lungo(200 caratteri)	note

I codici interni assegnati sono:

- *_HIDD* – Campo nascosto
- *_RDO* – Campo visibile, ma non modificabile (sola lettura)
- *_MOD* – Campo visibile e modificabile

Contenuto della tabella *TWF_Campi_K2Activity_ModalitaRappresentazione*

ID	<i>WF_Campi_K2Activity_ModalitaRappresentazione</i>	<i>CodiceInterno</i>	<i>Note</i>
1	Nascosto	<i>_HIDD</i>	Campo non visibile
2	Sola Lettura	<i>_RDO</i>	Campo visibile ma non modificabile
3	Modificabile	<i>_MOD</i>	Campo visibile e modificabile

10.2.6.5 Process Instance Folio

Ogni Istanza di Processo o *Process Instance* (Processo in esecuzione) ha una proprietà chiamata **Folio** che rappresenta un nome user-friendly per quella istanza di processo. Il **Folio** consente di identificare un'Istanza di Processo all'interno di K2 Service Manager. Di conseguenza bisogna stabilire un modo automatico di assegnazione del nome di *Folio* per ogni Istanza di Processo. Alcune possibilità:

- ❑ Nome processo + nome tabella IFG + id record
- ❑ Stringa fissa configurabile per process + valore di uno dei campi della tabella IFG per il record corrente (non necessariamente deve essere uno dei datafield del process)

11. S.U.A.P. - STRUTTURA DEL MODELLO WORKFLOW

11.1 Introduzione

La realizzazione di un diagramma di flusso (workflow) richiede competenze complesse che sappiano integrare le procedure già esistenti nell'Ente Pubblico con un processo di razionalizzazione ed ottimizzazione del flusso procedimentale. Ma chi opera già all'interno di un ente come responsabile del SUAP è sicuramente in grado di conoscere l'iter delle pratiche mentre non necessariamente ha competenze in materia di tecniche gestionali.

Con applicazioni accessibili via Web è possibile realizzare la mappa delle attività dei procedimenti. Il sistema è utilizzato sia dall'utente esterno, che può, quindi, inoltrare la propria domanda via Internet, controllare lo stato di avanzamento e essere informato via mail per eventuali richieste di integrazione con sospensione dei termini.

In questo capitolo si studierà un esempio pratico di applicazione del Sistema Workflow al Sistema di gestione dello Sportello Unico per le Attività Produttive (S.U.A.P.). Prima di tutto bisogna definire la struttura organizzativa del SUAP dopo di che studiare la sua implementazione come sistema di Workflow. In questo esempio ci sarà **interazione** tra i database (K2, K2log e K2SQLUM) di K2.net 2003 e i database (Database principale e quelli degli utenti) esterni in cui sono memorizzati tutti i dati del Portale internet del comune da dove saranno creati e gestiti i processi.

11.2 Definizione

Il sistema SUAP è un sistema informatizzato di **archiviazione, gestione e pubblicazione** delle **pratiche** secondo le attuali disposizioni normative.

Applicare un sistema Workflow al SUAP consentirà di gestire un flusso di informazioni, in rete telematica (Internet/Intranet), tra gli enti coinvolti nel processo con la creazione di un protocollo informatico che consentirà all'imprenditore di accedere all'informazioni sullo stato del procedimento.

11.3 Funzionalità

Con il sistema Workflow del SUAP è possibile :

- ❑ Supportare l'operatore di sportello dei vari comuni fornendo all'impresa informazioni sui procedimenti da attivare in seguito all'attivazione di un determinato intervento.
- ❑ Gestire automaticamente l'iter della pratica
- ❑ Tracciare e monitorare lo stato delle pratiche attivate
- ❑ Gestire gli avvisi sullo stato di avanzamento della pratica

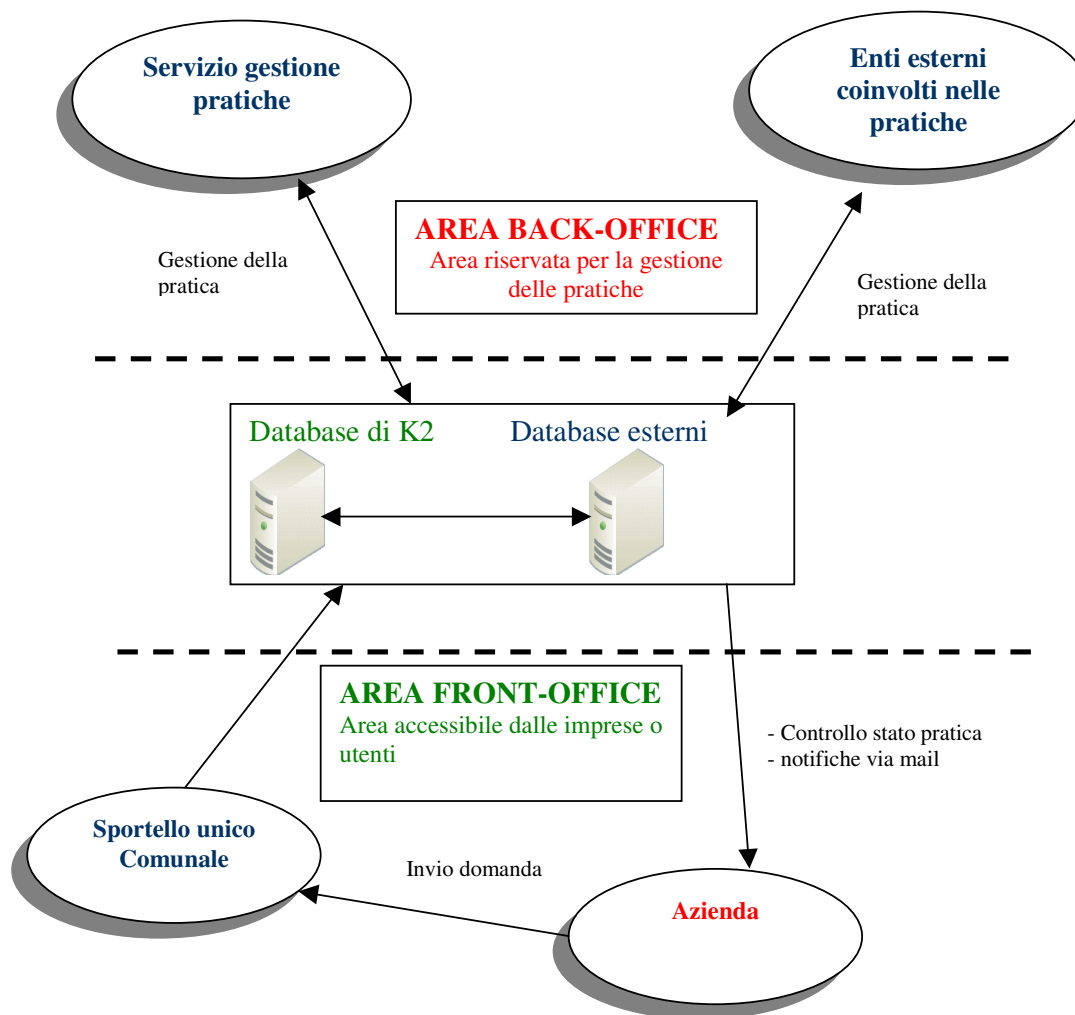


Figura 11.1: Attori coinvolti nel sistema Workflow SUAP

11.4 Funzionalità dell'area Back-office del Portale internet

Da quest'area gli amministratori del sistema, tramite quest'area riservata non visibile a tutti gli utenti, possono gestire sia le pratiche che tutti i software che pilotano il motore del Workflow. Prima di tutto l'amministratore deve definire la struttura (inserimento delle fasi, flussi...) del Workflow tramite il servizio *Settaggio Workflow*.

11.5 Funzionalità Front-office del Portale internet

Da quest'area tutti gli attori del Workflow (Ente, comune, Impresa ...) possono accedere per inserire, consultare, aggiornare le pratiche autenticandosi tramite una username ed una password, fornite dal sistema di gestione delle pratiche.



Figura 11.2: Accesso all'Area riservata lato utente del portale internet del comune

Dal Front-office (Area utente) è possibile:

- ⇒ Consultare e selezionare le procedure
- ⇒ Visualizzare e accedere alla modulistica

Presentare la pratica inserendo i dati anagrafici del titolare dell'intervento

11.6 Diagramma di flusso

Prima di procedere bisogna disegnare il diagramma di flusso(struttura) del sistema Workflow SUAP. In un secondo momento, come spiegato nei capitoli precedenti, è possibile

disegnare il diagramma workflow corrispondente direttamente da K2.net 2003 Studio ed esportare la soluzione sul server.

11.6.1 Diagramma di flusso originale

Di seguito c'è un esempio di diagramma di flusso del SUAP di una pubblica amministrazione locale (PAL) della Provincia di Modena.

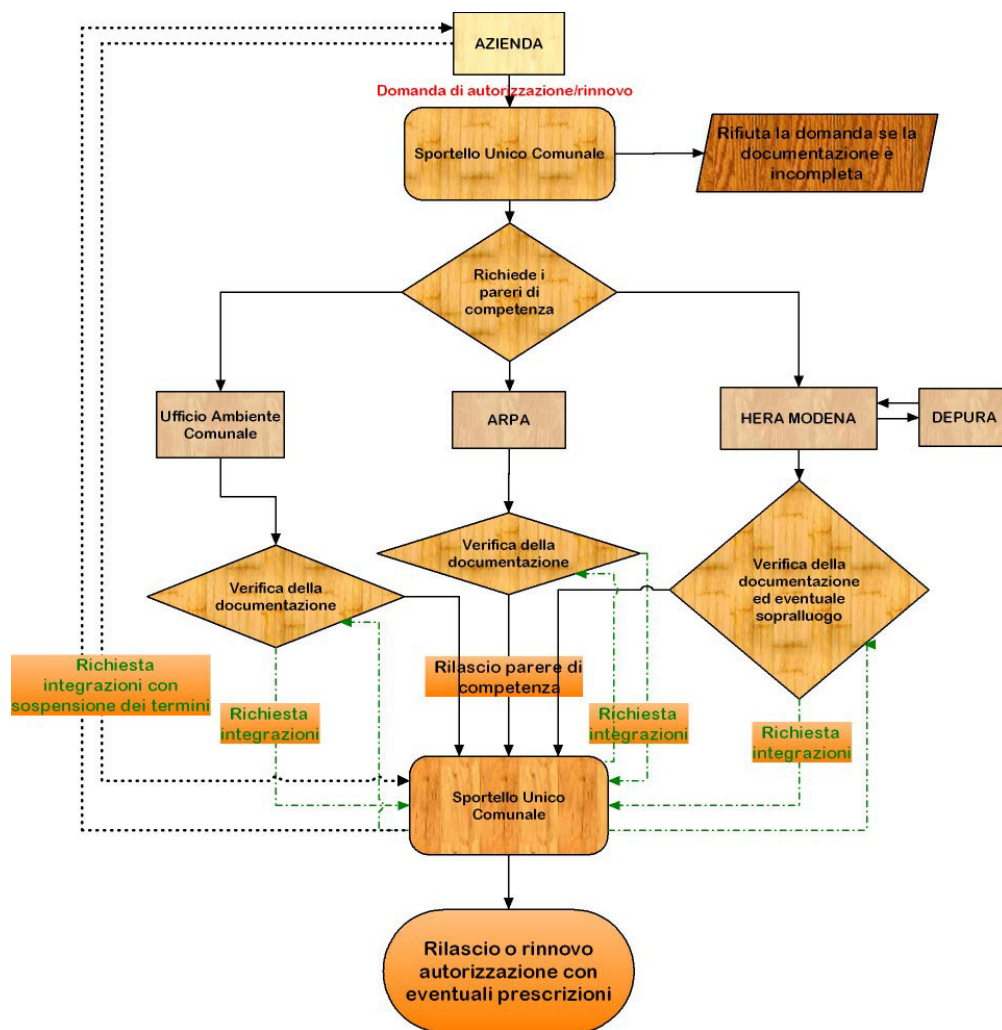


Figura 11.3: Diagramma di flusso del SUAP

11.6.2 Descrizione diagramma di flusso Suap

Come si vede da diagramma di sopra il flusso inizia con una domanda di autorizzazione o rinnovo. Dopo una prima verifica lo sportello unico può rifiutare la domanda per documentazione incompleta. Se la domanda risulta completa lo sportello unico richiede allora i pareri dell'Ufficio ambiente comunale , dell'Arpa e di Hera Modena che devono rilasciare pareri di competenza oppure richiedere l'integrazione. Lo sportello può di nuovo fare richiesta di integrazione agli uffici precedenti. In fine è possibile fare richiesta di integrazione direttamente all'azienda (che ha fatto domanda) con sospensione dei termini, oppure rilasciare/rinnovare l'autorizzazione.

11.6.3 Diagramma Workflow Suap disegnato con K2.net

La figura successiva rappresenta il corrispondente diagramma del SUAP modellato per il sistema workflow e disegnato con K2.net Studio

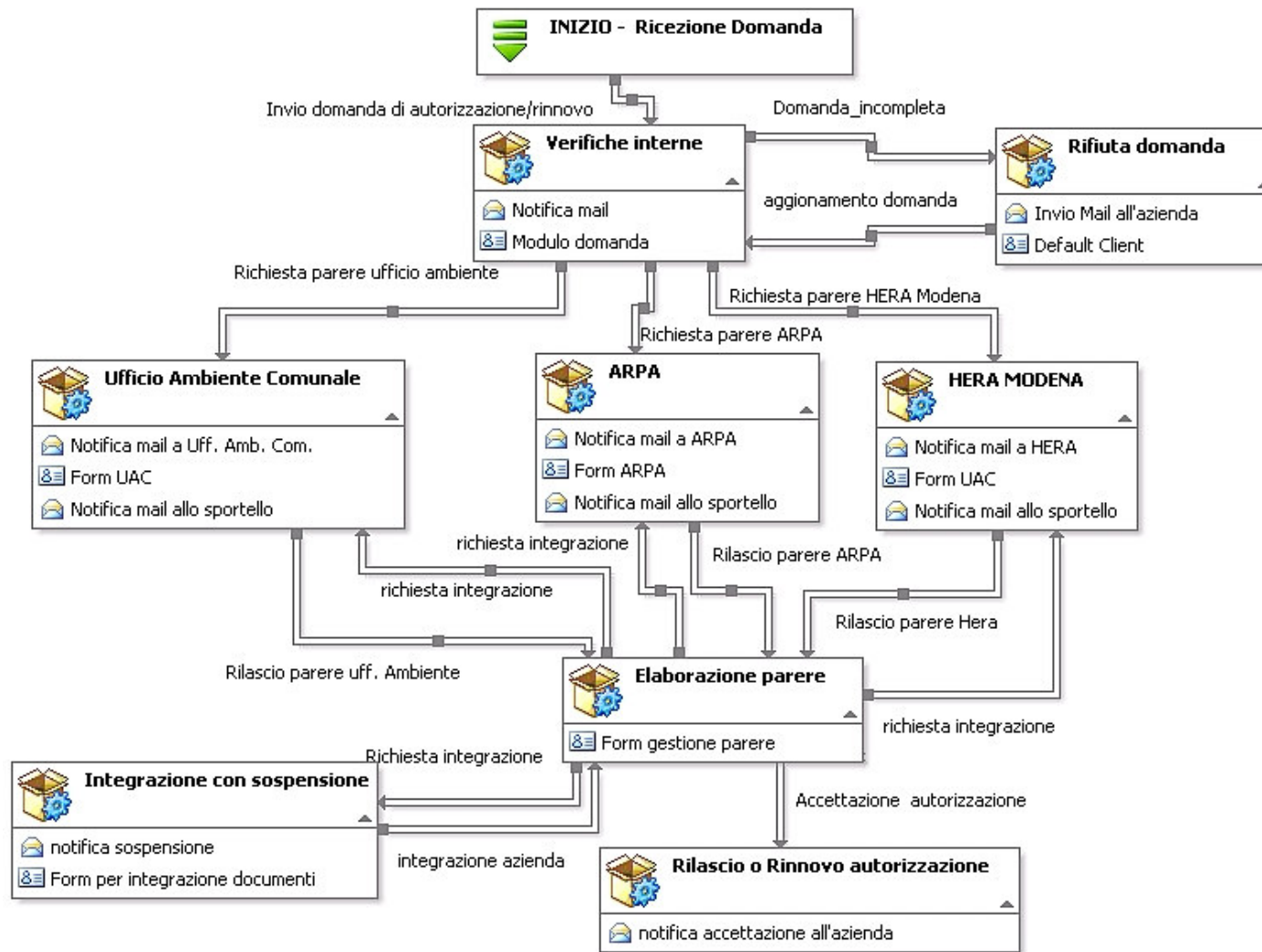


Figura 11.4: Diagramma Workflow del Suap rimodellato in K2.net Studio

11.6.3.1 Descrizione del diagramma Workflow Suap rimodellato

In questo diagramma di flusso rimodellato l'interazione interna alla struttura Hera con Depura non è stata presa in considerazione essendo visto come struttura unica.

11.6.3.2 Attori del processo di workflow

Come si vede dal diagramma l'ufficio "lo sportello unico comunale" è suddiviso in due strutture (aree) "Verifiche interne" e "Elaborazione parere".

I vari attori del workflow sono :

- ⇒ **Sportello unico comunale** (che riceve e gestisce le pratiche)
- ⇒ **Azienda**(che deve fare avviare il processo inviando una domanda)
- ⇒ **Ufficio Ambiente Comunale**(che deve verificare la documentazione e rilasciare un parere)
- ⇒ **ARPA**(che deve verificare la documentazione e rilasciare un parere)
- ⇒ **HERA MODENA**(che deve verificare la documentazione e rilasciare un parere)

Le attività dove sono richiesti gli interventi dell'uomo(chiamati *client event*: per esempio aggiornare i vari campo da un form) per fare proseguire il processo sono assegnati agli utenti o gruppi di utenti come illustrato nella seguente figura. Per semplicità verrà scritto solo l'username di prova di ogni utente per le varie attività.

11.6.3.3 Attività del processo di Workflow

Le varie attività del processo sono state definite rimodellando il diagramma originale per adattarlo al sistema di Workflow. Le varie attività sono:

- a) **Verifiche interna:** quest'Attività rappresenta l'utente(o utenti) che fa parte dello staff dello Sportello unico Comunale autorizzato a fare un primo controllo della pratica. Richiede agli enti coinvolti i pareri di competenza oppure respinge tutte le pratiche incomplete .
- b) **Rifiuta domanda:** quest'Attività gestita automaticamente dalla macchina (server) ha il compito di mandare una segnalazione

via mail all'azienda(il suo rappresentante o titolare) per segnalare il rifluito della domanda.

- c) **Ufficio Ambiente Comunale:** quest'Attività rappresenta l'utente (o utenti) autorizzato dell'Ufficio che deve verificare la documentazione e rilasciare un parere
- d) **ARPA:** quest'Attività rappresenta l'utente (o utenti) autorizzato dall'Arpa che deve verificare la documentazione e rilasciare un parere
- e) **HERA MODENA:** quest'Attività rappresenta l'utente (o utenti) autorizzato da Hera che deve verificare la documentazione e rilasciare un parere.
- f) **Elaborazione parere:** quest'Attività rappresenta l'utente(o utenti) che fa parte dello staff dello Sportello unico Comunale autorizzato a controllare i pareri di competenza ricevuti dai vari enti e procedere al rilascio/rinnovo delle autorizzazioni oppure richiedere l'integrazione all'aziende (che ha fatto domanda) con sospensione dei termini.
- g) **Integrazione con sospensione :** quest'Attività gestita automaticamente dalla macchina (server) ha il compito di mandare una segnalazione via mail all'azienda(il suo rappresentante o titolare) per segnalare la sospensione della pratica e consentire l'integrazione dal portale.
- h) **Rilascio o rinnovo autorizzazione :** quest'Attività gestita automaticamente dalla macchina (server) ha il compito di mandare una segnalazione via mail all'azienda(il suo rappresentante o titolare) per segnalare il rilascio/rinnovo delle autorizzazioni.

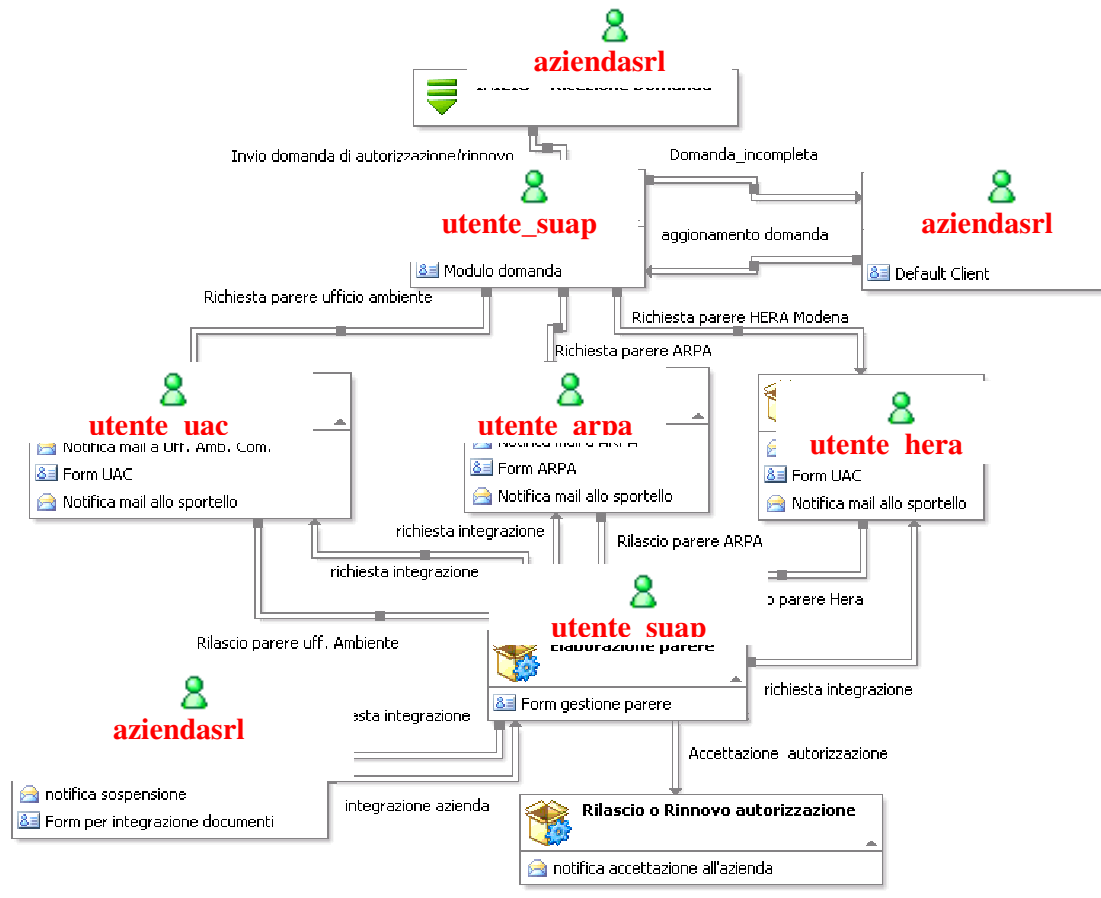


Figura 11.5: utenti (partecipanti) assegnati alle attività del processo

11.6.3.4 Eventi del processo di Workflow

Gli eventi sono principalmente gestiti dall'uomo attraverso i form via web e dalla macchina inviando per esempio le mail di notifiche.

⇒ **Form(Modulo)** : il form non è altro che un modulo virtuale dove gli utenti possono inserire/modificare o solo visualizzare i dati. Per esempio l'azienda per inoltre la sua pratica dovrà compilare un form dalla sua area riservata.



Figura 11.6: Simbolo dell'evento form da K2.net Studio

Nome *:	<input type="text"/>
Cognome *:	<input type="text"/>
Indirizzo:	<input type="text"/>
Data:	<input type="text" value="10/11/2006"/>

Figura 11.7: Esempio form

⇒ **Notifiche via mail:** Invece l'evento di invio mail è gestito automaticamente dal server di K2.net che invia la mail automaticamente al destinatario inserito durante il disegno del processo in k2.net studio

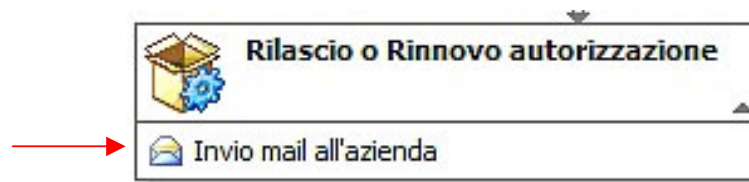


Figura 11.8: Simbolo dell'evento mail da K2.net Studio

11.6.4 Il deployment della soluzione disegnato in K2.net Studio

Un volta disegnato e impostate le proprietà della Solution ,Progetti , Processi , attività e eventi , è possibile fare il deploy sul server di test o di produzione. Come spiegato nel precedente capitolo prima di fare il deploy la soluzione viene compilata.

11.6.4.1 File e cartelle della *solution*

Durante la creazione della solution(con vari progetti,processi...) viene automaticamente creato i vari file e cartelle corrispondente. In questo caso è stato creato una cartella con il nome “*SUAP_TESI_BERTRAND*” . All'interno della cartella è stato creato contemporaneamente alla cartella il file *SUAP_TESI_BERTRAND.ksn* che rappresenta la solution. Nella cartella *SUAP_TESI_BERTRAND* c'è una sottocartelle di nome *Progetto1Suap* che rappresenta il nome del progetto creato.

Nome	Dimensione	Tipo	Data ultima modifica
Bin		Cartella di file	23/11/2006 18.50
Progetto1Suap.kpj	1 KB	File KPJ	23/11/2006 18.46
Richiesta Autorizzazione-Rinnovo.kpr	130 KB	File KPR	23/11/2006 18.46

Figura 11.9: Cartelle e file della solution generati da K2.net studio

11.6.4.2 Scelta del server di Produzione o test

Dopo la creazione del Progetto della Soluzione con K2.net studio è possibile scegliere il server dove sarà fatto il deploy dopo aver completato il disegno.

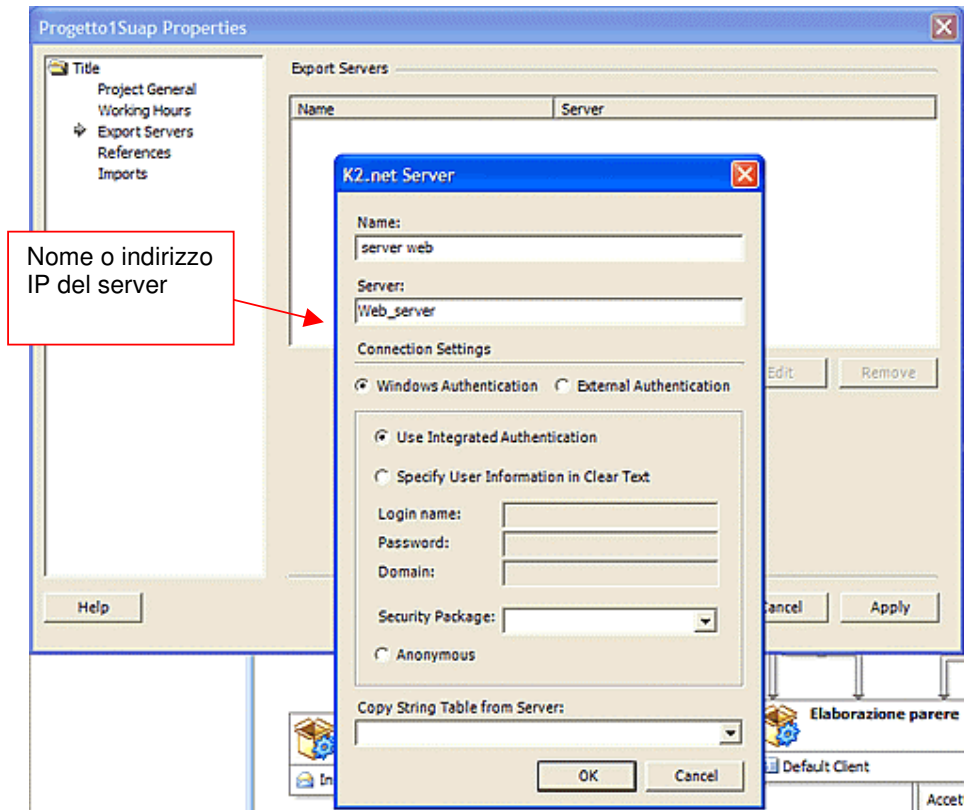


Figura 11.10: scelta del server dove esportare la soluzione

11.6.4.3 Compilazione del processo in K2.net studio

Come spiegato in precedenza, prima di fare il deploy della soluzione in un server, bisogna prima compilare il o i processi definiti nella soluzione. K2.net fa sempre la compilazione in modo automatico prima di procedere con l'esportazione. Di seguito c'è un esempio di compilazione del processo chiamato "Richiesta Autorizzazione-Rinnovo".

Compiling Richiesta Autorizzazione-Rinnovo...

Error - Activity Ufficio Ambiente Comunale has client event(s) and no destination rule

Error - Activity Verifiche interne has client event(s) and no destination rule

Error - Activity Elaborazione parere has client event(s) and no destination rule

Error - Activity HERA MODENA has client event(s) and no destination rule

Gli errori segnalati nella compilazione del processo sono dovuti al fatto che alle varie attività con eventi client (Eventi gestiti dall'uomo e non dalla macchina) non sono stati assegnati utenti per gestire tali eventi.

11.6.4.4 Oggetti di K2.net Studio usati per fare il deploy sul server

Prima di fare il deploy bisogna:

1. creare la solution con i vari progetti e processi con K2.net studio.
2. Scegliere un server di produzione o test dove esportare la solution come spiegato sopra.
3. Identificare il percorso assoluto della cartella della solution

Esempio di codice in C# per la gestione delle solution e fare il deploy con K2.studio:

```
public void CreateNewK2Solution()
{
    K2Studio.Application MyK2Application = new K2Studio.Application();
    K2Studio.Project MyK2Project;
    K2Studio.ProcessFolder MyK2ProcessFolder;
    K2Studio.Process MyK2Process;
    K2Studio.Processes MyK2Processes;
    int icount;

    //Crea una nuova Solution K2Studio
    //Serve per specificare l'estensione del file
    MyK2Application.Solution.Create("SUAP_TESI_BERTRAND ", @"C:\SUAP_TESI_BERTRAND \
SUAP_TESI_BERTRAND.ksn");

    // Crea il Progetto
    MyK2Project = MyK2Application.Solution.Projects.AddNew("Progetto1Suap ", @"C:\
SUAP_TESI_BERTRAND \ Progetto1Suap.kpj", K2Studio.EnumLanguage.VBasic);

    //Aggiunge un server di Esportazione
    MyK2Project.ExportServers.Add("K2Server");
    // Sistema di autenticazione al server k2
    MyK2Project.ExportServers.Item(0).WindowsAuthentication = true;
    MyK2Project.ExportServers.Item(0).Server = "K2Server";

    //aggiunge le References
    MyK2Project.References.Add("System.XML.dll", "", true);
    MyK2Project.References.Add("System.Web.dll", "", true);
    MyK2Project.References.Add("System.DirectoryServices.dll", "", true);
}
```

```

// Seleziona una cartella di un progetto esistente
MyK2ProcessFolder = MyK2Project.ProcessFolder;

//Aggiunge i Processo nella cartella dei Processi facendo un ciclo per ogni processo
MyK2Processes = MyK2ProcessFolder.Processes;
if(MyK2Processes.Count < 1)
{
    //Crea il Processo
    //Serve per specificare l'estensione del file
    MyK2Process = MyK2ProcessFolder.Processes.AddNew("MyK2Process", @"C:\K2Solutions\MyK2P
rocess.kpr");
    MyK2Process.Description = "My Test Process";
    MyK2Process.ExpectedDuration = 28800;
    MyK2Process.Priority = 1;
    MyK2Process.WorkingHours.UseParent = true;
    //Bind the Process Template
    MyK2Process.Template = MyK2Process.Application.Templates.Process.Item(0);
    MyK2Process.Open();

    //Crea il DataFields
    AddDataFields(MyK2Process);

    //Aggiunge le Attività
    AddActivities(MyK2Process);

    //Aggiunge le Linee
    AddLines(MyK2Process);

    //Salva le modifiche
    MyK2Application.Solution.Save();
    MyK2Application.Visible = true;

    //ESPORTA IL PROCESSO AL SERVER K2.net
    ExportProcess(MyK2Process);
}
else
{
    for(icontains = 1; icount <= MyK2Processes.Count; icount++)
    {
        //Find process in Collection
        MyK2Process = MyK2Processes.Item(icontains);
        if(MyK2Process.Name == "MyK2Process")
        {
            //Save Process
            MyK2Process.Save();
        }
        MyK2Process = null;
    }
}
MyK2Processes = null;
}

```

12. GESTIONE E STRUTTURA DEI DATABASE DI K2 E DEL PORTALE INTERNET SUAP

12.1 Gestione database utenti

Uno degli obiettivi di questa tesi è da fare condividere sullo stesso database centralizzato degli utenti di K2.net i vari utenti appartenenti a database esterni (relativi a vari portali internet o reti civiche). Di conseguenza non è stata presa in considerazione l'ambiente active directory di Windows dove sono recuperati di default tutti gli utenti workflow di K2.net. Di conseguenza è stato installato il componente K2.net SQLUM che crea automaticamente un database degli utenti K2SQLUM

12.2 Problemi riscontrati

Durante le prove eseguite ci sono stati problemi di sicurezza dei dati del workflow in quando tutti gli utenti (username, password...) memorizzati nel database principale SQLUM erano visti come appartenente allo stesso portale internet o stesso ente. Questo non è accettabile in quanto nel database principale devono convivere dati di portali diversi.

UserID	UserName	UserEmail	UserDescription	ManagerID	UserPassword
1	administrator	bertrand@progettidiimpresa.it	amministratore		adminpass
2	suap	suap@portale1.it	prova utente		suapass

Esempio contenuto tabella *K2Users* degli utenti nel database SQLUM

Nella tabella di sopra gli utenti "administrator" e "suap" saranno visibili, anche se non appartengono al Portale, in K2.net Studio, se è stato impostato come server K2.net il server dove è installato il database SQLUM. Quindi k2.net Studio fa vedere tutti gli utenti della tabella *K2Users* del database SQLUM

12.3 Risoluzione dei problemi riscontrati

Per risolvere il problema si può per esempio cambiare l'username degli utenti aggiungendo per esempio alla fine un nome identificativo del portale di appartenenza.

Esempio: l'username **suap** diventa **suap@portale1** di conseguenza la tabella di sopra diventa:

UserID	UserName	UserEmail	UserDescription	ManagerID	UserPassword
1	administrator	bertrand@progettidiimpresa.it	amministratore		adminpass
2	suap@portale1	suap@dominioportale1.it	prova utente		suapass
3	sportello@portale1	sportello@dominioportale1.it	prova utente		suapass

Esempio contenuto aggiornato tabella *K2Users* degli utenti nel database SQLUM

Dopo aver modificato l'username nella tabella principale del database SQLUM bisogna creare una database(esempio: portale1-SQLUM) con le stesse tabelle del database SQLUM con solo gli utenti del portale . Per esempio la tabella *K2users* del database portale1-SQLUM sarà:

UserID	UserName	UserEmail	UserDescription	ManagerID	UserPassword
1	suap@portale1	suap@ dominioportale1.it	prova utente		suapass
2	sportello@portale1	sportello@dominioportale1.it	prova utente		suapass

Esempio contenuto della tabella *K2Users* degli utenti nel database "Portale1-SQLUM"

Da notare che l'utente administrator non appartiene al portale di nome portale1 di conseguenza non sarà visibile in tale portale internet .

Nella figura di sotto viene illustrato l'interazione tra tutti i database citati sopra

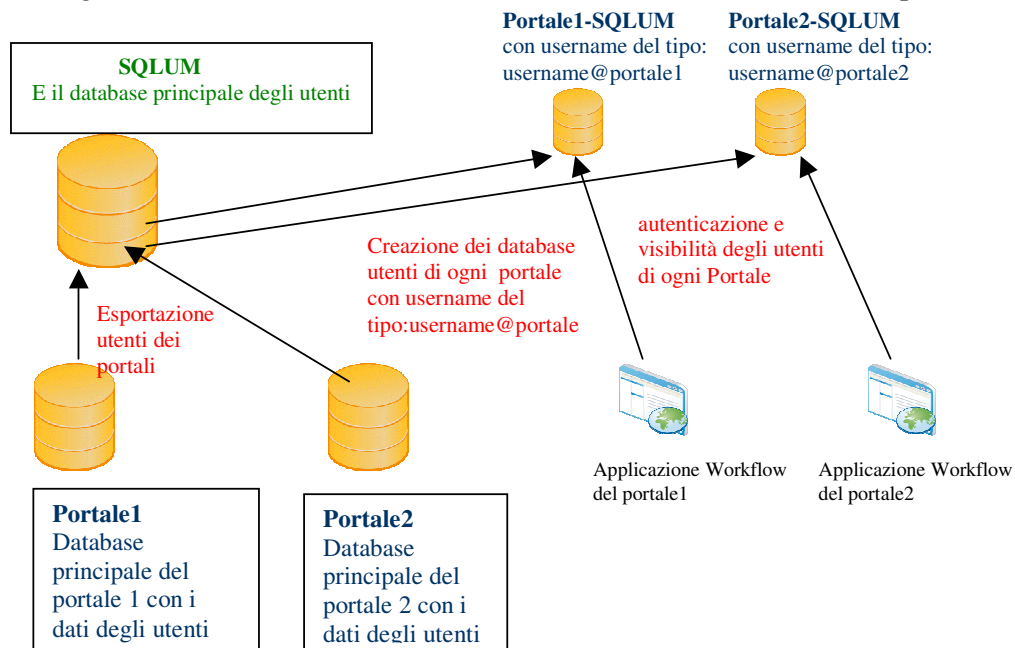


Figura 12.0.1: Interazione tra i database per la gestione degli utenti di ogni portale

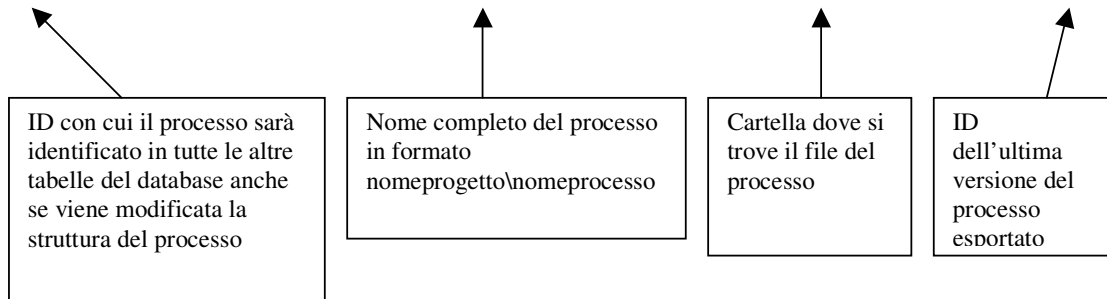
12.4 Architettura dei database per la gestione dei processi

12.4.1 Tabelle processo del database di K2

Le tabelle principali per la gestione dei processi , come spiegato nei capitoli precedenti sono *_ProcSet* (contiene il nome dei processi esportati sul server), *_Proc* (contiene tutte le varie versioni dei processi) e *_ProcData* (contiene tutti i campi definiti nel processo).

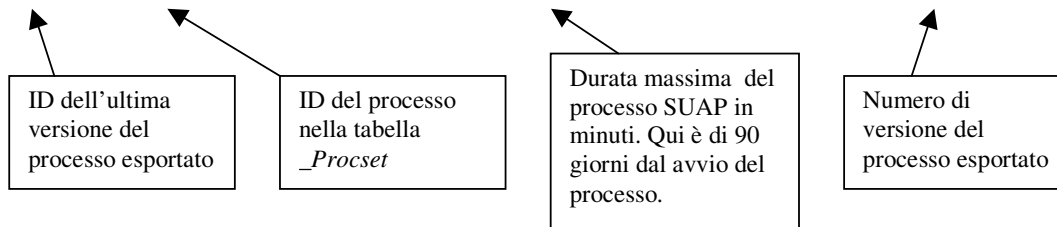
Tabella *_ProcSet*

ID	Name	FullName	Folder	Descr	ProcVerID
1	processo01	progetto\processo01	progetto		1
2	MAILMAN	LUCA\MAILMAN	LUCA		12
3	Richiesta Autorizzazione-Rinnovo	Progetto1Suap\Richiesta Autorizzazione-Rinnovo	Progetto1Suap		13



Alcuni campi della Tabella *_Proc*

ID	ProcSetID	ExportID	MetaData	Priority	ExpectedDuration	WorkID	LogLevel	Language	Ver	ChangeDate
1	1	1		0	0	0	3	1	1	26/10/2006
2	2	2		0	0	0	3	2	1	27/10/2006
3	2	3		0	0	0	3	2	2	30/10/2006
4	2	4		0	0	0	3	2	3	30/10/2006
5	2	5		0	0	0	3	2	4	30/10/2006
6	2	6		0	0	0	3	2	5	30/10/2006
7	2	7		0	0	0	3	2	6	30/10/2006
8	2	8		0	0	0	3	2	7	31/10/2006
9	2	9		0	0	0	3	2	8	31/10/2006
10	2	10		0	0	0	3	2	9	31/10/2006
11	2	11		0	0	0	3	2	10	31/10/2006
12	2	12		0	0	0	3	2	11	31/10/2006
13	3	13		0	129600	0	3	2	1	23/11/2006



12.4.2 Tabelle processo nel database del Portale internet

contenuto della tabella TWF_K2Process

ID	WF_K2Project	WF_K2Process
1		LUCAMAILMAN
2		Progetto1Suap\Richiesta Autorizzazione-Rinnovo

12.4.3 Struttura tabelle delle attività

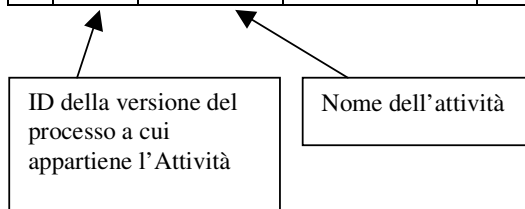
12.4.3.1 Tabelle delle attività del database di K2

La tabelle principale per la gestione delle Attività è *_Act* (contiene tutte proprietà di ogni attività).

Alcuni campi della Tabella *_Act*

ID	ProcID	Name	ExpectedDuration	WorkID	Slots	UseTran	PrecRuleID	DestRuleID	SucRuleID
18	8	Attivita01	0	0	1	FALSO	28	29	30
19	8	Start	0	0	1	FALSO	0	0	0
20	9	Approval	0	0	1	FALSO	32	33	34
21	9	Attivita01	0	0	1	FALSO	38	39	40
22	9	Start	0	0	1	FALSO	0	0	0
23	10	Approval	0	0	1	FALSO	42	43	44
24	10	Attivita01	0	0	1	FALSO	47	48	49
25	10	Start	0	0	1	FALSO	0	0	0
26	11	Attivita01	0	0	1	FALSO	51	52	53
27	11	Start	0	0	1	FALSO	0	0	0
28	12	Start	0	0	1	FALSO	0	0	0
29	12	Attivita01	0	0	1	FALSO	56	57	58
30	13	ARPA	0	0	1	FALSO	61	62	63
31	13	Rifiuta domanda	0	0	1	FALSO	0	0	0
32	13	Verifiche interne	0	0	1	FALSO	65	66	67
33	13	Elaborazione parere	0	0	1	FALSO	70	71	72
34	13	Integrazione con sospensione	0	0	1	FALSO	73	0	74
35	13	Rilascio o Rinnovo autorizzazione	0	0	1	FALSO	0	0	0
36	13	Ufficio Ambiente Comunale	0	0	1	FALSO	77	78	79

37	13	HERA MODENA	0	0	1	FALSO	81	82	83
38	13	INIZIO - Ricezione Domanda	0	0	1	FALSO	0	0	0



12.4.3.2 Tabelle delle attività nel database del Portale

Tutte le attività del processo sono definiti una tabella del database del Portale internet.

Contenuto della tabella *TWF_K2Activity*

ID	WF_K2Process	WF_K2Activity	Startup
1	1	Approval	FALSO
2	1	Kick-off	VERO
3	2	INIZIO - Ricezione Domanda	VERO
4	2	ARPA	FALSO
5	2	Rifiuta domanda	FALSO
6	2	Verifiche interne	FALSO
7	2	Elaborazione parere	FALSO
8	2	Integrazione con sospensione	FALSO
9	2	Rilascio o Rinnovo autorizzazione	FALSO
10	2	Ufficio Ambiente Comunale	FALSO
11	2	HERA MODENA	FALSO

12.5 Le variabili del processo

I vari dati del processo sono memorizzati in campi nel database di K2.net e nel database del portale de Portale internet del comune. Questi campi del processo, definiti prima in K2.net Studio come variabili, sono visibili o nascosto a seconda dello stato in cui si trova la pratica. Queste variabili saranno anche definiti, usando il software IFG(Input Form Generator), nel database (che chiamerò da adesso in poi DB_PORTALE) del portale internet.

Nome campo	Tipo campo	Descrizione
Nome	string	Nome dell'utente (di un azienda) che inoltra la pratica
Cognome	string	Cognome dell'utente (di un azienda) che inoltra la

		pratica
email	string	E-mail dell'utente (di un'azienda) che inoltra la pratica
nomeAzienda	string	Nome o ragione sociale dell'azienda che inoltra la pratica
piva	string	Partita Iva dell'azienda per la quale si fa la domanda
tipoDomanda	string	Variabile di tipo drop down list visibile all'utente azienda consente di selezionare il tipo di domanda. I suoi valori sono: -autorizzazione -rinnovo autorizzazione
parere_arpa	string	Variabile che consente di tipo drop down list consente di sapere se l'ARPA ha inviato il suo parere che può essere: -SI (risposta positiva) -NO (risposta negativa) -NULL (non ha risposto)
parere_UAC	string	Variabile che consente di tipo drop down list consente di sapere se l'Ufficio Ambiente Comunale ha inviato il suo parere che può essere: -SI (risposta positiva) -NO (risposta negativa) -NULL (non ha risposto)
parere_Hera	string	Variabile che consente di tipo drop down list consente di sapere se HERA ha inviato il suo parere che può essere: -SI (risposta positiva) -NO (risposta negativa) -NULL (non ha risposto)
Note_UAC	string	Variabile che consente di memorizzare il parere di competenza dell'Ufficio Ambiente Comunale
Note_ARPA	string	Variabile che consente di memorizzare il parere di competenza dell'ARPA
Note_HERA	string	Variabile che consente di memorizzare il parere di competenza di HERA
gestione_pratica	string	Variabile di tipo drop down list consente allo sportello comunale accettare la domanda oppure richiedere l'integrazione all'azienda richiedente o agli uffici per gestioni pareri . I suoi valori sono: -integrazioneazienda -integrazioneaufficio -rilascio_rinnovo
Stato_documentazione	string	Variabile di tipo drop down list visibile nell'attività "verifiche interne" consente di sapere se la documentazione è completa o non. I suoi valori sono: -completa -incompleta
note_prescrizione	string	Variabile che consente allo sportello unico di

		comunicare eventuali prescrizioni durante il rilascio/rinnovo dell'autorizzazione.
note_verifiche	string	serve per inserire la nota che verrà inserita nella mail inviata all'azienda in caso di rifiuto per documentazione incompleta.

Elenco delle variabili usati nel processo Suap definiti sia nei database di K2 e del Portale

12.6 Creazione dei form per la varie attività del Processo dal Portale

Dopo il disegno e l'esportazione del processo sul server di K2. Bisogna a questo punto creare i form che serve di interfaccia con i vari attori del workflow (utente dell'azienda, dello sportello, dell'ARPA ...). La creazione del form si fa direttamente dall'area riservata del Portale internate tramite il software *Input Form Generator* come illustrato nella figura di sotto.

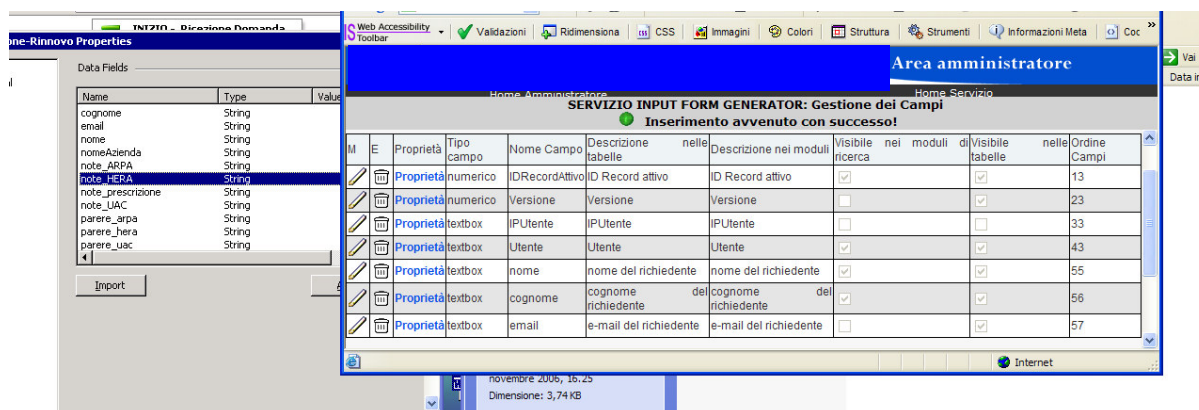


Figura 12.0.2: Creazione dei vari campi del processo tramite Input Form Generator

In seguito alla creazione del form generale con tutti i campi (variabili del processo definiti in K2.net Studio), viene creata automaticamente una tabella che chiamo TAB_FORM1. Di seguito c'è lo script di creazione della tabella.

```
CREATE TABLE [utenteweb].[TAB_FORM1] (
  [ID] [int] IDENTITY (1, 1) NOT NULL ,
  [datainserimento] [smalldatetime] NOT NULL ,
  [nome] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
  [cognome] [nvarchar] (50) COLLATE Latin1_General_CI_AS NULL ,
  [email] [nvarchar] (100) COLLATE Latin1_General_CI_AS NULL ,
  [nomeazienda] [nvarchar] (150) COLLATE Latin1_General_CI_AS NULL ,
  [piva] [nvarchar] (20) COLLATE Latin1_General_CI_AS NULL ,
```

```

[TAB_FORM1_tipoDomanda] [int] NULL ,
[TAB_FORM1_stato_documentazione] [int] NULL ,
[TAB_FORM1_parere_arpa] [int] NULL ,
[TAB_FORM1_parere_hera] [int] NULL ,
[TAB_FORM1_parere_uac] [int] NULL ,
[TAB_FORM1_gestione_pratica] [int] NULL ,
[note_ARPA] [nvarchar] (300) COLLATE Latin1_General_CI_AS NULL ,
[note_HERA] [nvarchar] (300) COLLATE Latin1_General_CI_AS NULL ,
[note_UAC] [nvarchar] (300) COLLATE Latin1_General_CI_AS NULL ,
[note_verifiche] [nvarchar] (300) COLLATE Latin1_General_CI_AS ,
[note_prescrizione] [nvarchar] (300) COLLATE Latin1_General_CI_AS
NULL
) ON [PRIMARY]

```

Sono campi elenco con i relativi valori definiti in tabelle correlate con il nome del campo

Elenco di alcune Tabelle correlate dei vari campi elenchi definiti nella tabella *TAB_FORM1* :

Alcuni campi della tabella *TAB_FORM1_stato_documentazione*

ID	datainserimento	stato_documentazione
1	23/11/2006	
2	23/11/2006	completa
3	23/11/2006	incompleta

Il corrispondente script:

```

CREATE TABLE [utenteweb].[TAB_FORM1_stato_documentazione] (
    [ID] [int] IDENTITY (1, 1) NOT NULL ,
    [datainserimento] [smalldatetime] NOT NULL ,
    [stato_documentazione] [nvarchar] (100) COLLATE Latin1_General_CI_AS NOT
NULL ,
    ...
) ON [PRIMARY]
GO

```

Alcuni campi della tabella *TAB_FORM1_tipoDomanda*

ID	datainserimento	tipoDomanda
1	23/11/2006	
2	23/11/2006	domanda di autorizzazione
3	23/11/2006	domanda di rinnovo


E possibile definire anche dei campi non presenti nel processo definito da K2.net Studio però gestito esternamente dal portale internet. Per esempio sono stati definiti vari

campi(allegato_uac, allegato_arpa) per consentire ai vari attori (utente dell'Arpa, dell'ufficio ambiente comunale) di allegare dei documenti.

12.6.1 Visibilità dei campi per ogni attività



Una volta creato il form con tutti campi necessari per l'esecuzione del processo a questo punto bisogna per ogni attività nascondere o rendere visibili i campi a seconda dell'attività.

Variabili visibili e modificabili dall'azienda per avviare il processo

	Nome	Cognome	Email	nomeAzienda	piva	tipoDomanda
 INIZIO - Ricezione Domanda	visibile	visibile	Visibile	visibile	visibile	visibile
ARPA	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Rifiuta domanda	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Verifiche interne	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Elaborazione parere	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Integrazione con sospensione	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Rilascio o Rinnovo autorizzazione	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto
Ufficio Ambiente Comunale	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
HERA MODENA	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura

Le variabili che servono per avviare la pratica sono visibili e modificabili all'attività *INIZIO*

Variabili visibili e modificabili dallo sportello unico comunale

	gestione_pratica	Stato_documentazione	note_prescrizione	note_verifiche
INIZIO - Ricezione Domanda	Nascosto	Nascosto	Nascosto	Nascosto
ARPA	Nascosto	Nascosto	Nascosto	Nascosto
Rifiuta domanda	Nascosto	Nascosto	Nascosto	Nascosto
 Verifiche interne	Nascosto	visibile	Nascosto	Nascosto
 Elaborazione parere	visibile	visibile-sola lettura	visibile	visibile

Integrazione con sospensione	Nascosto	Nascosto	Nascosto	Nascosto
Rilascio o Rinnovo autorizzazione	Nascosto	Nascosto	Nascosto	Nascosto
Ufficio Ambiente Comunale	Nascosto	Nascosto	Nascosto	Nascosto
HERA MODENA	Nascosto	Nascosto	Nascosto	Nascosto

La variabile *Stato_documentazione* e visibile sono nell'attività *Verifiche interne* che serve per verificare che il documento sia completo, dopo non è più modificabile nelle attività successive.

Variabili visibili agli enti per il parere

	parere_arpa	parere_UAC	parere_Hera	Note_UAC	Note_ARPA	Note_HERA
INIZIO - Ricezione Domanda	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto
→ ARPA	visibile	Nascosto	Nascosto	Nascosto	visibile	Nascosto
Rifiuta domanda	Nascosto	Nascosto	Nascosto	Nascosto	visibile	Nascosto
Verifiche interne	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto
Elaborazione parere	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura	visibile-sola lettura
Integrazione con sospensione	Nascosto	Nascosto	Nascosto	Nascosto	visibile-sola lettura	Nascosto
Rilascio o Rinnovo autorizzazione	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto	Nascosto
→ Ufficio Ambiente Comunale	Nascosto	visibile	Nascosto	visibile	Nascosto	Nascosto
→ HERA MODENA	Nascosto	Nascosto	visibile	Nascosto	Nascosto	visibile

13. SIMULAZIONE DEL SISTEMA WOKFLOW SUAP

13.1 Introduzione

In questo ultimo capitolo si fa un test per verificare la funzionalità e correggere eventuali errori che si possono verificare durante l'esecuzione del processo. Per fare questo si usano gli strumenti disponibili in *K2.net 2003* come il *Workspace o Service Manager*.

13.2 Gestione degli utenti usati durante il test

13.2.1 Utenti

Durante il test sono stati usati cinque utenti assegnati alle varie attività che richiedono l'intervento dell'uomo per essere considerati dal processo workflow come completati. Questi utenti sono registrati nel portale con opportuno username e password. Di seguito ci sono i cinque utenti identificati ognuno con il suo username di accesso al portale e l'attività su cui dovrà intervenire:

Attività	Utente	Da fare
INIZIO - Ricezione Domanda	aziendasrl	Utente azienda che deve avviare il processo compilando il form iniziale
Verifiche interne	suap_prova	Utente dello sportello unico comunale che dovrà controllare se i dati immessi sono completi.
Rifiuta domanda	aziendasrl	Utente azienda che deve aggiornare il form iniziale in caso di domanda incompleta.
ARPA	utente_arpa	Utente dell'Arpa che deve dare il suo parere o richiedere eventuali integrazioni
Ufficio Ambiente Comunale	utente_uac	Utente dell'Ufficio Ambiente Comunale che deve dare il suo parere o richiedere eventuali integrazioni
HERA MODENA	utente_hera	Utente di Hera che deve dare il suo parere o richiedere eventuali integrazioni
Elaborazione parere	suap_prova	Utente dello sportello unico comunale che, a questo stadio, deve controllare i vari

		pareri degli enti coinvolti , dare l'autorizzazione o chiedere eventuale integrazione all'azienda
Integrazione con sospensione	aziendasrl	Utente azienda che deve integrare alcuni documenti ,richiesti dallo sportello unico comunale, alla sua pratica
Rilascio o Rinnovo autorizzazione	Nessun utente	Attività gestita dal server

Utenti assegnati alle varie Attività

13.2.2 Permessi degli utenti

Ogni utente o gruppi di utenti del sistema workflow del Suap deve ha alcuni permessi a seconda delle azioni che deve compiere ad ogni attività. Per esempio l'utente(o gruppo) identificato come azienda può avviare un processo, invece l'utente(o gruppo) identificato dello Sportello ha il permesso non solo di avviare i processi ma anche di monitorarli.

User Name	Admin	Start	View	View Partici...	Server Event
administrator	*	*	*	*	*
suap_prova@c...	*	*	*	*	*
Anonymous		*	*	*	
aziendasrl@ca...		*	*	*	
utente_uac@c...		*	*	*	
utente_hera@...		*	*	*	
utente_arpa@...		*	*	*	

Figura 13.1: permessi agli utenti del processo

Come si vede dalla figura precedente tutti gli utenti tranne suap_prova e administrator, possono avviare e vedere le istanze del processo. Invece gli utenti suap_prova e administrator hanno i permessi di amministrazione oltre a quelli standard.

13.2.3 Accesso area riservata

Tutti gli utenti devono accedere all'area riservata del front office (lato utente) del Portale internet per avviare o vedere le istanze del processo. Solo l'utente amministratore può accedere all'area riservata del Back office.

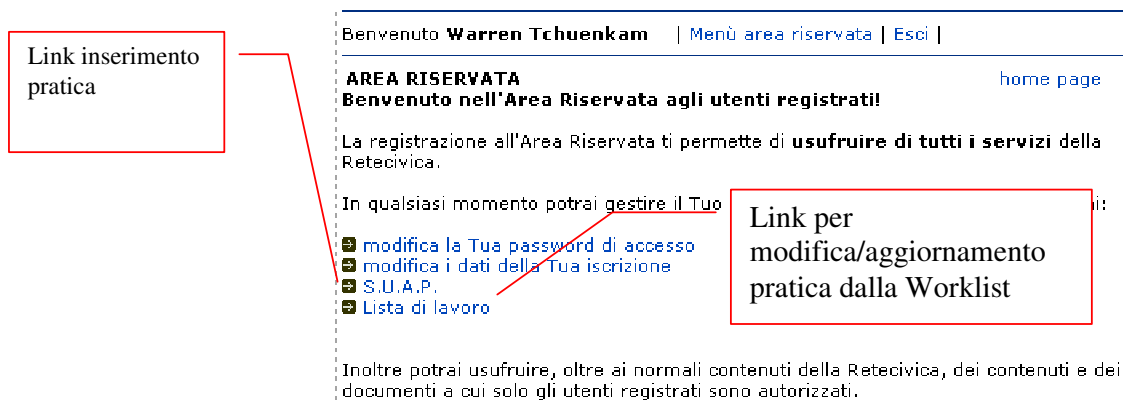


Figura 13.2:Menu area riservata

13.3 Avvio e Gestione del processo

13.3.1 Avvio del processo

Come illustrato dal diagramma di flusso del processo è l'utente azienda (nel nostro esempio usiamo *azienda srl*) che deve avviare il processo compilando ed inviando il form(modulo) della pratica. Tutti i dati della pratica sono immediatamente inoltrati all'utente (in questo caso l'utente *suap_prova*) dello sportello unico appena l'utente a completato il modulo. Visto che per questo caso di studio la firma digitale non è stata presa in considerazione, l'utente(dall'area riservata del Portale) dopo aver compilato il form relativo alla sua pratica di richiesta/rinnovo autorizzazione dovrà stamparla e consegnare il documento in formato cartaceo firmato allo Sportello Unico Comunale.

Figura 13.3:Alcuni campi del modulo per inserire una nuova pratica

13.3.2 Ricezione pratica

Appena un nuovo utente fa una nuova domanda (nuova istanza di processo) il sistema invia automaticamente una mail di notifica all'utente *suap_prova* come illustrato dalla figura seguente.

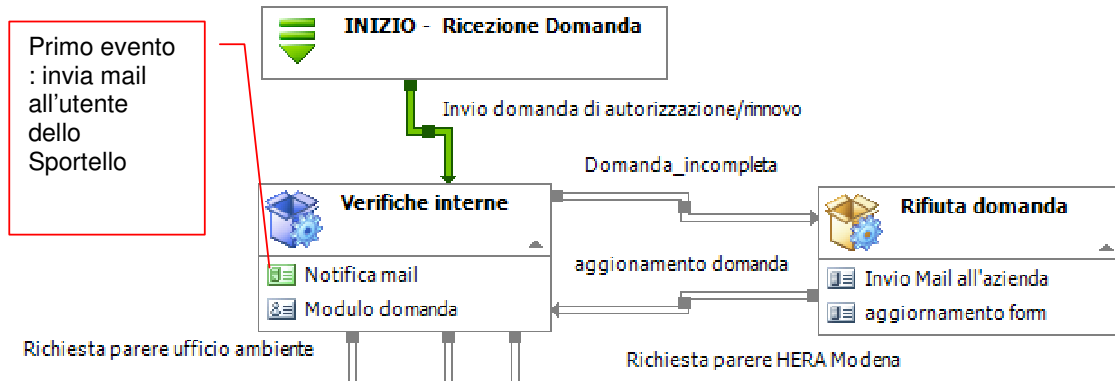


Figura 13.4: Stato del processo visto dal Workspace dopo l'invio di una nuova pratica

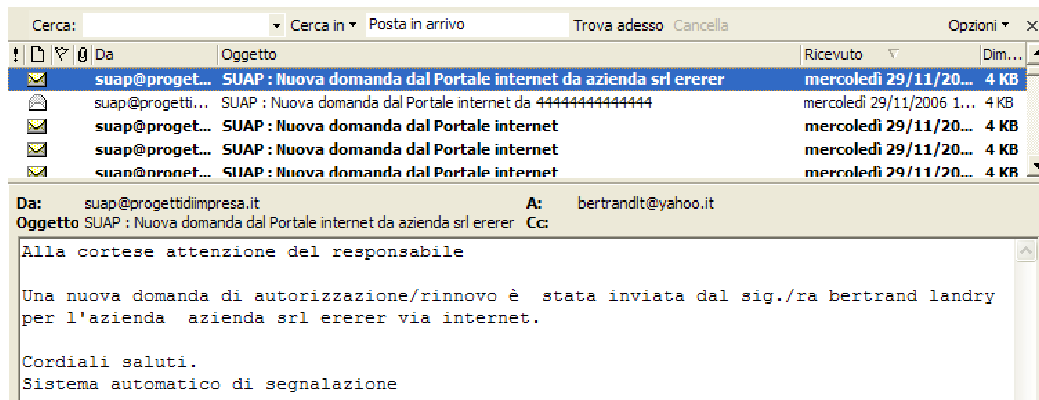


Figura 13.5: Mail di notifica Allo sportello unico dopo l'inserimento di una nuova pratica

13.3.3 Gestione Pratica

Dopo l'accesso all'area riservata l'utente dello sportello (in questo caso utente *suap_prova*) dovrà dall'area riservata controllare la pratica ed inoltrare agli enti competenti la pratica o respingere la domanda se risulta incompleta.

Processo	Attività	Folio	Link
Richiesta Autorizzazione-Rinnovo	Elaborazione parere	Progetto1Suap\Richiesta Autorizzazione-Rinnovo	Vai
Richiesta Autorizzazione-Rinnovo	Elaborazione parere	Progetto1Suap\Richiesta Autorizzazione-Rinnovo	Vai
Richiesta Autorizzazione-Rinnovo	Elaborazione parere	Progetto1Suap\Richiesta Autorizzazione-Rinnovo	Vai

Figura 13.6: Worklist dell'utente dello sportello unico comunale

13.3.4 Gestione delle istanze del processo

Ogni volta che viene inserita una nuova pratica viene creata automaticamente una nuova istanza del processo. Tutti i dati relativi alla gestione delle istanze del processo sono memorizzati nel database K2log gestito da K2.net 2003

a) Monitoraggio delle istanza

Oltre a sapere lo stato delle pratiche è possibile monitorare il flusso del processo dal Workspace di K2.net 2003

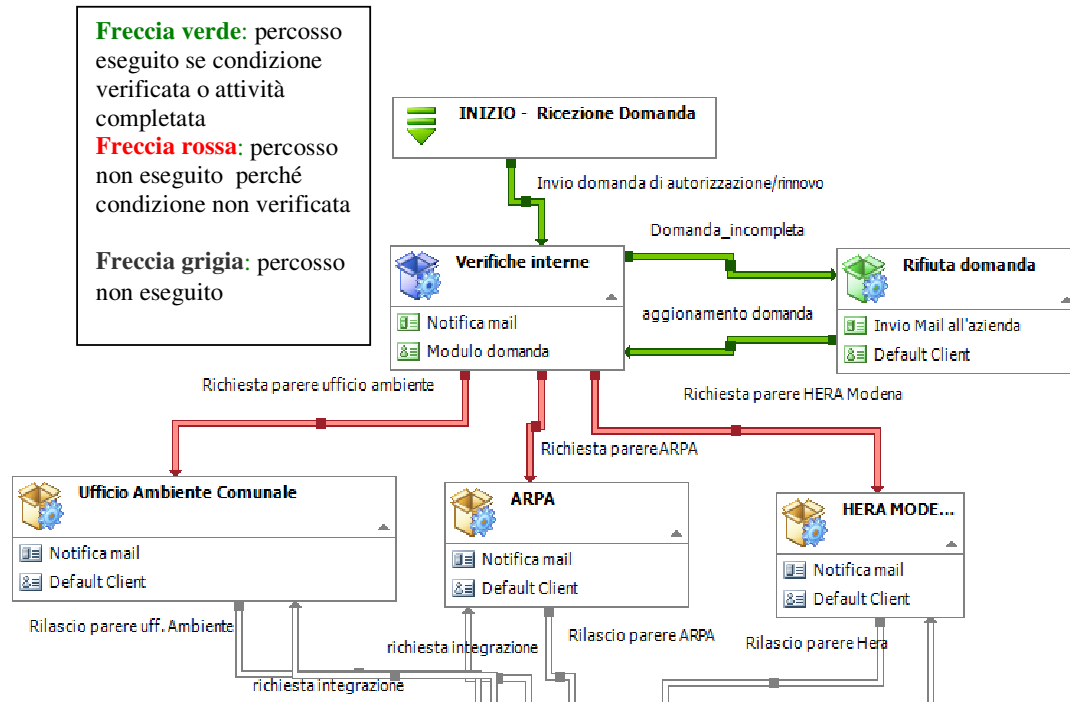


Figura 13.7: Monitoraggio del flusso del processo dalla Workspace

14. CONCLUSIONI

Questo progetto è stato molto stimolante ed importante per la mia carriera professionale e universitaria. Abbiamo visto come l'integrazione di un'applicazione di workflow con applicazioni esterne sia possibile ed offre all'utente finale un'interfaccia web molto semplice e chiaro anche per utenti meno esperti. Lo studio, l'implementazione e l'esecuzione della gestione automatica della pratiche dello sportello unico alle imprese(SUAP) consente di capire quanto è importante per le aziende, la pubblica amministrazione e soprattutto per il cittadino uno sportello automatico perché consente di ottimizzare i tempi di gestione della pratiche, di informare (via mail) con tempestività gli utenti sullo stato delle pratiche da loro inviate o gestite.

La tecnologia workflow è attualmente in grado di supportare in modo adeguato le esigenze di integrazione, migliorandone le prestazioni (efficacia) e velocizzandone le comunicazioni (efficienza). Le prestazioni sono più positive in quanto questa tecnologia riesce a proporre meccanismi di coordinamento e di collaborazione sostituendosi all'attività umana. Come visto le applicazioni workflow consentono di realizzare sia una raccolta che un opportuno smistamento delle informazioni, inoltre facilitano l'ottimizzazione di un processo – soprattutto produttivo(o di gestione delle procedure burocratiche per la pubblica amministrazione), ove le componenti in gioco sono molte - agevolando le decisioni di sequenzializzazione o parallelizzazione dei compiti da svolgere.

La ricerca organizzativa è necessaria poiché la gestione della tecnologia workflow ha un grosso impatto sulle operazioni, sulle strategie e sulle analisi operative dell'Azienda.

15. BIBLIOGRAFIA

Per la stesura di questa tesi di laurea è stato utilizzato in prevalenza il manuale d'uso del software K2.net .2003 e documentazione on-line, elencati di seguito.

Libri

[1] “Developing enterprise Workflow Solutions Using K2.net 2003” Student Manual V.3.0
Copyright 2004 by SourceCode Technology Holding, Inc.

Nota: il nome K2.net 2003 appartiene alla società SourceCode Technology Holding, Inc.

I nomi dei vari prodotti richiamati in questa tesi appartengono ai rispettivi proprietari.

Documentazione on-line

K2.net | Enterprise Workflow for .NET

<http://www.k2workflow.com/>

Workflow management

http://it.wikipedia.org/wiki/Workflow_management

The Workflow Management Coalition

<http://www.wfmc.org/>

“Collaborare bene... produrre meglio” di *Gian Franco Stucchi*

<http://www.cbritaly.it/Sezioni/Made-for-Italy/Collaborare-bene.-produrre-meglio>

“L' azienda ideale e la fabbrica ideale delle idee” di *Dott. Marcello Sabatini*

<http://www.msconsulting.it/ideale1.htm>

I sistemi di Workflow

http://www.i-dome.com/docs/pagina.phtml?_id_articolo=6187

Open Source Java XPDL workflow

<http://www.enhydra.org/workflow/shark/index.html>

OSWorkflow - OSWorkflow Overview

<http://www.opensymphony.com/osworkflow/>

Business Process Modeling Tools - Easy and Intelligent Business Process Modeler

<http://www.visual-paradigm.com/product/bpva/>

16. APPENDICE

16.1 Requisiti software per l'installazione

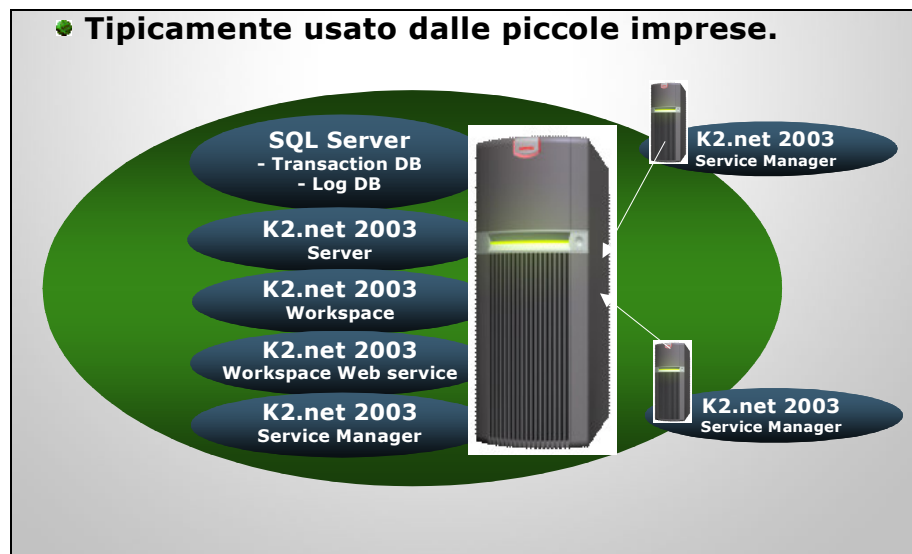
Prima di procedere alla procedura di installazione , di seguito ci sono i requisiti software :

- ⇒ Microsoft Windows 2003 Server (lato server)
- ⇒ Internet Information Service 6.0 con ASP.NET (lato server)
- ⇒ Microsoft SQL Server 2000 con Service Pack 3 o superiore oppure Microsoft Database Engine 2000 with Service Pack 3 (lato server)
- ⇒ Microsoft Active Directory or a compatible directory service(anche se non indispensabile)
- ⇒ Microsoft .NET Framework 1.1
- ⇒ Microsoft SharePoint Products and Technologies
- ⇒ Server SMTP (anche esterno per l'invio delle mail)
- ⇒ Visual Studio.NET (VB, NET or C#) (lato server o/e client)
- ⇒ Microsoft Windows XP(per i programmatori)
- ⇒ Microsoft Office XP (lato client)

16.2 Architetture hardware

L'architettura di K2.net 2003 e flessibile. E possibile configurazione l'architettura per tre livelli di utilizzo : Small, Medium(usato della medie imprese) , Large (usato della grandi imprese), Cluster (consente di creare un server logico composta da vari server fisici)

K2.net 2003 Architecture – Small



L'architettura di K2.net 2003 consente di installare sullo **stesso server** (questo per le piccole aziende) tutti questi componenti :

- ⇒ K2.net 2003 Database, incluso i database K2 e K2Log
- ⇒ K2.net 2003 SQLUM (database degli utenti da installare solo se non è installato l'Active Directory oppure non si vuole usare gli utenti dell'Active Directory)
- ⇒ K2.net 2003 Server
- ⇒ K2ROM Runtime .NET assemblies
- ⇒ K2.net 2003 Workspace
- ⇒ K2.net 2003 Management components
- ⇒ K2.net 2003 Documentation (Manuale)

K2.net 2003 Architecture – Medium



Il piano d'azione suggerito per una piccola o media impresa consiste nel dedicare un secondo server per dati dei vari componenti. Di conseguenza il server principale non dovrà eseguire SQL server liberando di conseguenza le risorse del server

K2.net 2003 data store è composto da due database chiamati “transaction runtime database”(K2.mdf) e K2.net log database(K2log.mdf). I due database possono essere installati sullo stesso server o su due server nella stessa rete LAN per ottimizzare le prestazioni. Microsoft SQL Server possono anche essere installati su piattaforma Cluster.

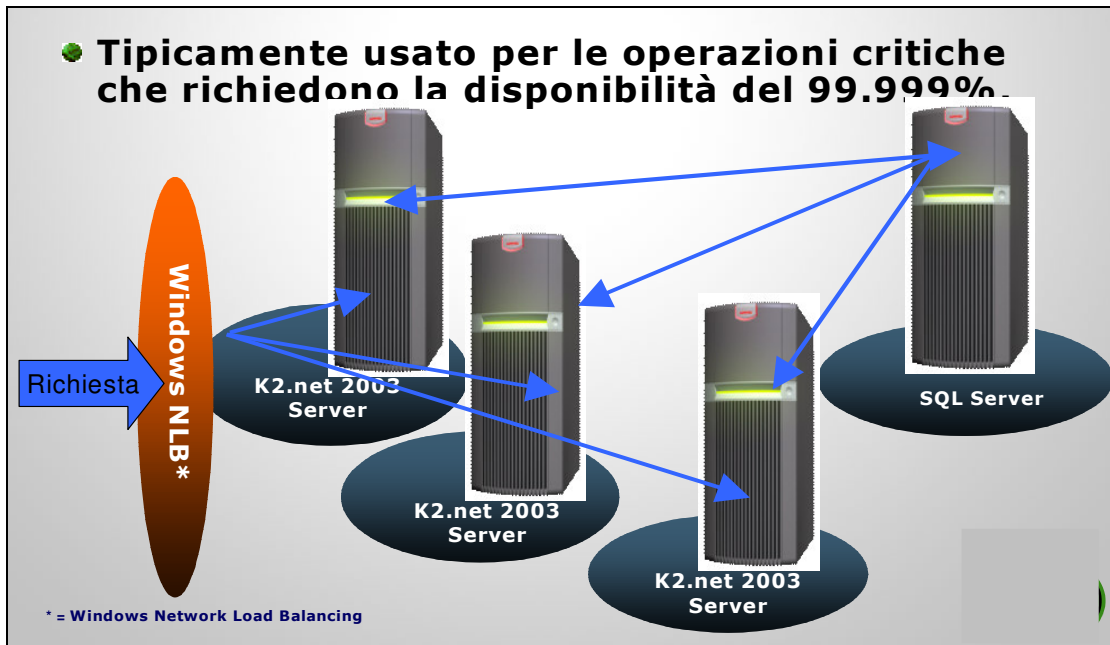
K2.net 2003 Architecture – Large



K2.net 2003 Server, K2.net 2003 Data store e K2.net 2003 Workspace possono essere eseguiti in ogni server dedicato separatamente. In queste condizioni tutte le risorse sono disponibili per l'esecuzione dei processi. Il secondo SQL server sarà dedicato ai dati delle funzioni e il terzo server sarà dedicato alla pubblicazione delle pagine web agli utenti finali.

Tutti i riferimenti di assembly .NET, che contengono riferimento al web service e assembly aggiunti come riferimento a K2.net 2003 solution in K2.net 2003 Studio, sono automaticamente esportati su K2.net 2003 Server da the K2.net 2003 Studio

K2.net 2003 Architecture – Clustered



K2.net Enterprise Workflow Server consente di creare un server logico composto da più server fisici.

Ogni nodo abbina la sua potenza di calcolo per creare una sinergia che consente di completare i lavori in k2.net Data Store come una singola unità ottimizzata e lo stato è mantenuto tra server che consente di eseguire le transazioni parallele .

In alternativa i server cluster multipli possono essere creati regionalmente ed usato per eseguire processi per regione.

I server cluster multipli consentono un accesso sicuro tra loro tramite K2.net Inter-Process Communication, per chiamare processi attraverso LAN, WAN o l'Internet, se è quando richiesto.

16.3 Installazione di K2.net 2003

In questo paragrafo sarà illustrato la procedura di installazione di K2.net 2003 in una rete priva di active directory , cioè gli utenti autorizzati ad accedere a K2.net saranno memorizzati in un database separato (dopo l'installazione di K2.net 2003 SQL User Manager). Questo è necessario nel caso in cui si vuole personalizzare l'autenticazione a k2.net a gruppi di utenti diversi.

Requisiti per l'installazione di K2.net 2003 SQL User Manager

Componente di installazione	Descrizione	Requisiti
K2.net 2003 SQL User Manager Server Components	Installa e configura K2.net 2003 Server per usare SQL User Manager. Questi componenti sono installati sulla macchina(Server) <u>dove è installato K2.net 2003 Server.</u>	K2.net 2003 SP3 o SP4
K2.net 2003 SQL User Manager Studio Components	Installa e configura K2.net 2003 Studio Templates per usare SQL User Manager. Questi componenti sono installati sulla macchina(Server o workstation) <u>dove è installato K2.net 2003 Studio.</u>	K2.net 2003 Studio SP3
K2.net 2003 SQL User Manager Database	Installa e configura il database di K2.net 2003 SQL User Manager. Questi componenti sono installati sulla macchina(Server) <u>dove è installato K2.net 2003 Server.</u>	Microsoft SQL Server 2005 o superiore; Microsoft SQL Server 2000 con Service Pack 3 o superiore; O Microsoft Database Engine 2000 con Service Pack 3 o superiore

Procedura di installazione/configurazione

Installazione K2.net 2003 SQL User Manager Sul server

- disinstallare le versioni precedenti di K2 (compresi eventuali plugins e database se non ci sono dati sensibili) se presenti
- installare K2.net 2003 SP3 o SP4 (tutti i componenti standard sul server:K2Server, Workspace, K2Studio). Da notare che in fase di installazione verrà creato una cartella virtuale K2V3 sotto un dominio configurato su IIS, oppure sotto il dominio predefinito (default web site)
- installare tutti i componenti di K2.net 2003 SQL User Manager (SQLUM Database;SQLUM Server components;SQLUM Studio components). Verrà creato un database chiamato SQLUM che può essere anche cambiato in fase di installazione.

Installazione di K2.net 2003 SQL User Manager su PC dei programmatori:

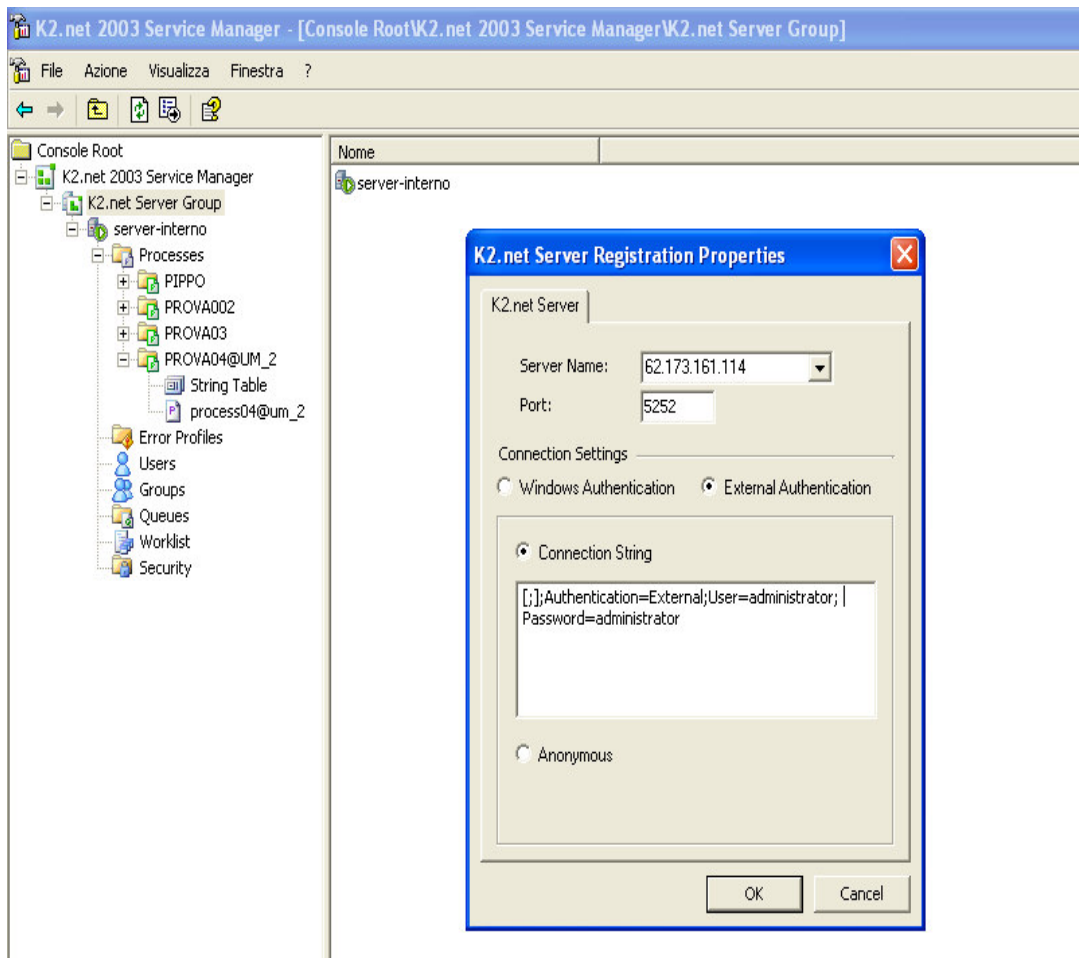
- installare K2.net 2003 Studio e K2.net 2003 service Manager

- installare solo SQLUM Studio Component (Questo permette a K2.net Studio di collegarsi agli utenti nel database SQLUM invece di quelli dell'active directory. disattivare SQLUM database)

TEST DA FARE DOPO L'INSTALLAZIONE:

⇒ **Accesso al server da K2.net 2003 Service Manager**

- 1- Dalla console del Server, fermare il servizio "K2.net 2003 Server " se avviato
- 2- Attivare(avviare) la console di "K2.net 2003 Server "
- 3- Aggiungere un utente di test nella tabella "K2Users" del database SQLUM (esempio :
username=administrator Password= passadmin)
- 4- Aprire K2.net 2003 Service Manager e creare una nuova registrazione "K2.net Server Registration"
- 5- Dalla proprietà di K2.net Server Registration:
 - Inserire nome del server
 - Attivare Extrenal authentication
 - Inserire nella connection string ([;];Authentication=External; User= administrator;
Password= passadmin)
 -



Stringa di connessione al server da K2.net 2003 Service Manager
 ⇒ **Accesso al workspace**

Per accedere al workspace bisogna digitare questo URL da un qualsiasi Browser
<http://nomeserver/K2V3/Workspace/Login.htm> dove “nomeserver” è il nome³, dominio⁴
 oppure indirizzo IP⁵ del server dove sono installati i componenti server di K2.net 2003.

> **Dalla Lan:** Basta inserire username e password dell'utente presente nel database K2SQLUM

> **Da internet:** Oltre ad inserire username e password dell'utente presente nel database K2SQLUM bisogna inserire username e password di un utente di windows solo perché durante l'installazione la cartella virtuale K2V3 non è abilitata all'accesso anonimo da internet in quanto richiede l'autenticazione Kerberos. Comunque questo aumenta il livello di sicurezza.

³ solo ci troviamo all'interno di una LAN(reti locale)

⁴ il dominio sotto il quale è stata creata la cartella virtuale in fase di installazione di K2.net 2003

⁵ indirizzo IP pubblico (per l'accesso da internet) o privato per l'accesso dalla LAN

Note : Se durante il login si verifica un errore del tipo:

“The request failed with HTTP status 401: Unauthorized”

allora bisogna fare queste modifiche sul server (dove è installato il K2.net 2003 server):

- 1- Dare il permesso di scrittura alla cartella “C:\WINDOWS\Temp” all’utente “everyone” o al gruppo di utenti (autorizzati)
- 2- Dare il permesso di lettura alla cartella “C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\Temporary ASP.NET Files” all’utente “everyone” o al gruppo di utenti (autorizzati)
- 3- Dare il diritto di lettura alla cartella “C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322\Temporary ASP.NET Files” all’utente “everyone” o al gruppo di utenti (autorizzati)

⇒ ACCESSO UTENTE DA K2.net 2003 Studio

Stringa connessione al database degli utenti dalla rete internet :

Si può modificare manualmente la stringa di connessione Per collegare a database degli utenti k2SQLUM. Basta andare nel percorso di installazione di K2.net 2003 Studio:
in questo caso

C:\Programmi\K2.net 2003\Bin , aprire il file **K2Studio.config** e cercare la voce

```
...
<DestinationRule>
  <DataSources>
    <DataSource Path="DEFAULT" Type="ActiveDirectory" />
  </DataSources>
  <SQLUM>
    <SQLUMSource
SQLUMConstring="Server=IPSERVERK2;UID=sa;PWD=;DataBase=K2SQLUM;" />
    </SQLUM>
  </DestinationRule>
...
```

Dove:

Server = nome o indirizzo IP del server

UID = nome dell’utente autorizzato ad accedere al database

Password = Password nome dell’utente autorizzato ad accedere al database

DataBase= Nome database degli utenti di quel portale

RINGRAZIAMENTI

Un sentito ringraziamento va a mia moglie, a mio figlio, ai miei genitori, a tutti i miei familiari (in particolare quelli di mia moglie) e amici, che, con il loro affetto e continuo sostegno morale, mi hanno permesso di raggiungere questo traguardo.

Desidero ringraziare la prof. Sonia Bergamaschi, l'ing. Lorenzo Canali e tutti quelli che mi hanno aiutato durante la stesura di questa tesi.

Dedico questa tesi a tutti i miei cari familiari defunti... Una dedica particolare al mio defunto padre che è sempre stato per me un riferimento.