



# **CMDBuild**

## ***Open Source Configuration and Management Database Overview Document***

**Versione 0.7.0  
Novembre 2007**

No part of this document may be reproduced, in whole or in part, without the express written permission of Tecnoteca s.r.l.

CMDBuild leverages many great technologies from the open source community:  
PostgreSQL, Apache, Tomcat, Eclipse, JasperReports, IReport  
We are thankful for the great contributions that led to the creation of that products

CMDBuild è un progetto realizzato e gestito da:

Comune di Udine – Servizio Sistemi Informativi e Telematici



Tecnoteca S.r.l. ([www.tecnoteca.com](http://www.tecnoteca.com))



Cogitek S.r.l. ([www.cogitek.it](http://www.cogitek.it))



CMDBuild è rilasciato con licenza GPL ([www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html))

Il sito ufficiale di CMDBuild è <http://www.cmdbuild.org>

## Sommario

Introduzione.....	5
Cos'è un CMDB.....	5
Perchè utilizzare un CMDB.....	6
Criteri di attuazione.....	6
Open Source.....	7
Le risposte di CMDBuild.....	8
Inventario dei beni informatici.....	8
Manutenzione del parco hardware.....	8
Gestione delle licenze software.....	8
Catalogo dei servizi.....	8
Tracciamento delle informazioni nel tempo.....	8
Gestione degli apparati di telefonia.....	9
Gestione della documentazione.....	9
Gestione dei processi.....	9
Caratteristiche di CMDBuild.....	11
Flessibilità.....	11
Object Oriented.....	11
Specializzazione.....	12
Interamente utilizzabile via web.....	12
Interoperabilità.....	13
Funzionalità di CMDBuild.....	14
Principali funzionalità del modulo Schema.....	14
Principali funzionalità del modulo Gestione Dati .....	15
Funzionalità base .....	15
Funzionalità avanzate.....	16
Motore di workflow.....	17
OCS Inventory .....	18
DCL (Double Choco Latte) .....	19
Tecnologia di CMDBuild.....	21
Ambienti di sviluppo e produzione:.....	21
Approfondimento: utilizzo avanzato di PostgreSQL.....	21
APPENDICE A: Glossario.....	23
ALLEGATO .....	23
ATTIVITA' .....	23
ATTRIBUTO .....	23
CI .....	23
CLASSE.....	23
CONFIGURAZIONE .....	24
DATABASE.....	24
DOMINIO.....	24
ITIL.....	24
LOOKUP.....	24
PROCESSO.....	25
RELAZIONE.....	25
REPORT.....	25
SCHEMA.....	25
SUPERCLASSE.....	25
TIPO DI ATTRIBUTO.....	26

APPENDICE B: Le Best Practice ITIL.....	27
APPENDICE C: Link utili.....	28
SALPA 2005.....	28
Conferenza Annuale 2005 di itSMF Italia .....	28
COMPACT 2006.....	28
Conferenza Annuale 2006 di itSMF Italia .....	28
Think Open 2007 .....	28
Linux day 2007 .....	28
ALTRA PA.....	28
Il sito ufficiale ITIL .....	29
ITIL IT Service Management Zone .....	29
L'Associazione itSMF Italia .....	29
PostgreSQL (database).....	29
Tomcat.....	29
Alfresco (repository documenti).....	29
Enhydra Shark (motore workflow) .....	29
OCS Inventory (Automatic Inventory).....	29

# Introduzione

CMDBuild è una applicazione Open Source per la configurazione e gestione del database della configurazione (CMDB) degli oggetti e servizi informatici in uso presso il Dipartimento IT di una organizzazione.

Gestire un Database della Configurazione significa mantenere aggiornata e disponibile per gli altri processi la base dati relativa agli elementi informatici utilizzati, alle loro relazioni ed alle loro modifiche nel tempo.

CMDBuild si ispira alle "best practice" ITIL (Information Technology Infrastructure Library), ormai affermatosi come "standard de facto", non proprietario, per la gestione dei servizi informatici secondo criteri orientati ai processi.

Con CMDBuild l'amministratore del sistema può costruire autonomamente il proprio CMDB (da cui il nome del progetto), grazie ad una apposita applicazione di configurazione che consente di aggiungere progressivamente nel sistema nuove classi di oggetti, nuovi attributi e nuove tipologie di relazioni.

E' possibile allegare documenti, immagini ed altri file alle schede archiviate in CMDBuild ed operare ricerche full text sui documenti di testo (tramite integrazione con il diffuso sistema open source di Gestione Documenti Alfresco).

Tramite il sistema implementato per la gestione del workflow è poi possibile definire in modo visuale con un editor esterno nuovi processi operanti sulle classi trattate nel sistema, importarli in CMDBuild ed eseguirli secondo i criteri indicati.

E' infine disponibile un sistema di interfaccia con fonti dati esterne (database e server mail) per l'aggiornamento dati da altri sistemi, utilizzato in particolare per aggiornare l'inventario automatico dell'hardware rilevabile in rete (tramite integrazione con il sistema open source OCS Inventory).

## Cos'è un CMDB

Un CMDB è un sistema di archiviazione e consultazione delle informazioni che descrivono e riguardano i sistemi informatici presenti in una organizzazione.

E' l'archivio centrale ed ufficiale che fornisce una visione coerente dei servizi IT.

E' un sistema dinamico che deve rappresentare costantemente lo stato di fatto e di conoscenza del patrimonio informatico e delle entità connesse.

E' un sistema di controllo dei processi svolti, descritti e gestiti tramite le funzionalità del workflow.



Quali elementi informatici gestisce il CMDB ?

- hardware: computer, periferiche, sistemi di rete, apparati di telefonia
- software: di base, di ambiente, applicativo
- documenti: progetti, contratti, manualistica
- altre risorse, interne ed esterne

A quali domande risponde il CMDB:

- dove si trova un CI (configuration item) ?
- chi lo usa ?
- di cosa fa parte ?
- da cosa è composto ?
- quali sono e dove si trovano altri CI analoghi ?
- ho licenze sufficienti per l'utilizzo del software ?
- cosa è successo nella vita del CI ?
- su quali altri CI impatta una eventuale modifica ?
- quali sono le attività in attesa di un mio intervento ?

## **Perchè utilizzare un CMDB**

Utilizzare un CMDB consente di avere sempre sotto completo controllo la situazione degli elementi informatici utilizzati, conoscendone in ogni momento la composizione, la dislocazione e le relazioni funzionali.

Informazioni mancanti o non aggiornate significano costi inutili, operazioni ridondanti, ritardo nella risoluzione dei problemi, intralcio alle attività aziendali.

Le parole chiave di un CMDB sono velocità di risposta e controllo del sistema

Un CMDB consente di:

- ridurre i problemi al proprio sistema informatico
- risolvere più velocemente i problemi residui
- risolvere al primo livello di risposta una maggiore percentuale di problemi, evitando di coinvolgere troppo frequentemente il personale più esperto

In altre parole diminuzione dei costi e miglioramento della qualità dei servizi.

## **Criteri di attuazione**

Se da un lato per organizzazioni di media ed elevata dimensione è assolutamente consigliabile adottare uno strumento di gestione del CMDB, dall'altro vanno attentamente valutate le modalità e le tempistiche di attuazione.

Nella gestione del progetto è importante:

- scegliere correttamente il livello di dettaglio con cui partire, che dovrà essere adeguato alle necessità dell'organizzazione ed alle risorse umane, finanziarie, informative e tecnologiche disponibili (se si è scelto un sistema realmente flessibile e modulare è molto meglio avvalersi della possibilità di successive estensioni autonome della struttura dati di supporto)

- adottare un sistema di gestione flessibile ed espandibile, che garantisca la possibilità di estendere secondo necessità il numero degli oggetti ed il livello informativo da gestire
- inserire il sistema informatico all'interno di un sistema organizzativo basato su procedure, ruoli e responsabilità

Un progetto di successo non può prescindere dalla corretta valutazione dell'impatto organizzativo del sistema nell'organizzazione e dalla esplicita e pubblica approvazione del management aziendale.

Avviare e gestire il progetto con la consulenza di esperti ITIL significa poi avvalersi dell'esperienza e delle "best practice" nate e collaudate in realtà diverse di paesi diversi, velocizzandone l'attuazione e riducendo i rischi di fallimento.

## **Open Source**

Open source è meglio!

Una soluzione open source consente di:

- evitare costi anche elevati di licenza
- disporre del codice sorgente che evita ogni dipendenza dal fornitore
- scegliere un prodotto che può evolvere con il contributo tecnico o di esperienza degli utilizzatori

## Le risposte di CMDBuild

Obiettivi di CMDBuild sono quelli di rispondere alle domande ed alle esigenze del Dipartimento IT, semplificarne le attività di gestione, garantire la disponibilità di informazioni sempre precise ed aggiornate.

CMDBuild garantisce risposte utili per la gestione dell'inventario, ma anche della manutenzione o della gestione delle licenze.

Di seguito alcune domande ricorrenti nella gestione di infrastrutture IT.

### Inventario dei beni informatici

- di quanti PC dispone il mio centro ?
- quali sono stati acquistati da uno specifico fornitore ?
- quanti PC nuovi ho installato annualmente negli ultimi tre anni ? e quanti ne ho dimessi ?
- quanti monitor ho in magazzino ? di che modello ? quando scade la garanzia ?

### Manutenzione del parco hardware

- dovendo trasferire un intero ufficio in un altro edificio su quanti oggetti IT devo intervenire ? qual'è la lista aggiornata ?
- fra i PC degli utenti che hanno richiesto l'acquisto di un particolare software quanti hanno meno di 256 MB di RAM e devono essere aggiornati ?
- e quanti di questi hanno più di 4 anni e sono fuori garanzia e potrebbe quindi essere il caso di dismettere ?
- quali server divengono irraggiungibili dovendo spegnere per manutenzione uno specifico gruppo di continuità ?
- quanti utenti condividono una determinata stampante ?

### Gestione delle licenze software

- ho licenze sufficienti per far utilizzare un determinato software a tutti gli utenti dell'Ufficio Tecnico ?
- qual è la lista delle licenze in uso presso ciascun dipartimento ?
- quali licenze sono in scadenza ?

### Catalogo dei servizi

- quali utenti devo preavvisare in caso di indisponibilità di una applicazione web ?
- di quali permessi dispone sull'applicazione gestionale l'utente con cui sono al telefono ?
- quali SLA devo garantire ?

### Tracciamento delle informazioni nel tempo

- ci sono uffici che lamentano numeri eccessivi di guasti ?

- su che server girava una data applicazione lo scorso anno ?
- quali utenti erano in possesso di un account per l'accesso alla rete ad una data specifica ?
- in che data ho installato l'ultima patch ad un determinato software ? che versione era ?

### **Gestione degli apparati di telefonia**

- quali utenti non hanno in dotazione un apparato telefonico ?
- ci sono prese telefoniche libere in una stanza dove è necessario spostare del personale ?

### **Gestione della documentazione**

- ho evidenza delle SLA da rispettare in relazione ad uno specifico servizio ?
- sono immediatamente recuperabili i documenti di progetto relativi ad una applicazione software commissionata ?

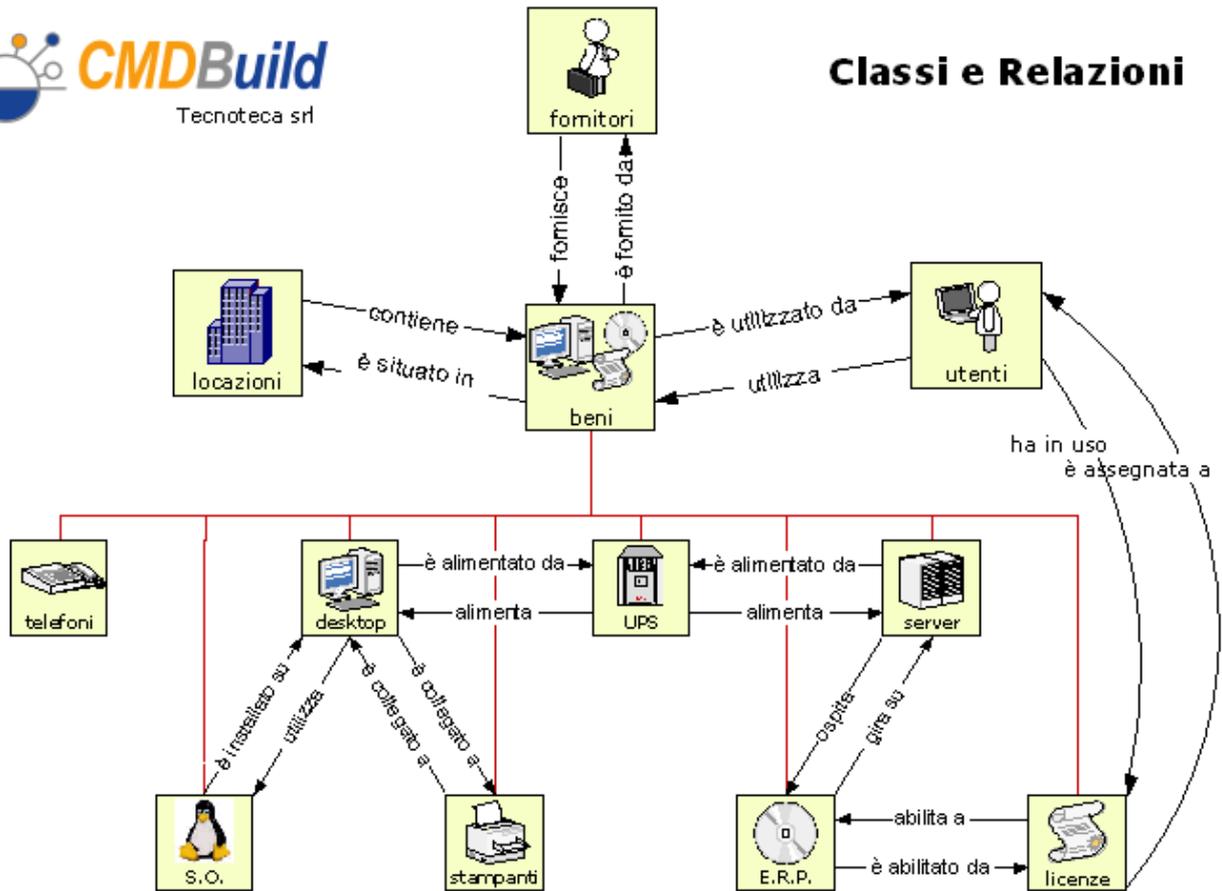
### **Gestione dei processi**

- ho organizzato le attività secondo processi standardizzati e idonei a garantire la massima efficacia ed efficienza ?
- ho adottato i criteri suggeriti da ITIL ?
- gli operatori sono guidati dal sistema nell'esecuzione delle attività previste ?

- quali sono le attività in attesa di un mio intervento ?



## Classi e Relazioni



## Caratteristiche di CMDBuild

CMDBuild vuole essere una soluzione robusta, estendibile e personalizzabile per i problemi di gestione delle infrastrutture IT:

- robusta, perché si basa su una architettura modulare e su “fondamenta” progettate per supportare estensioni della struttura dati e delle funzionalità di gestione
- estendibile, perché grazie ad una apposita applicazione di configurazione l'amministratore del sistema è in grado di aggiungere progressivamente nel sistema nuove classi di oggetti e nuove tipologie di relazioni da controllare, ritrovandosi un sistema sempre coerente con la struttura dati da modellare
- personalizzabile, perché in quanto Open Source è sempre possibile intervenire sul progetto per integrarne funzionalità mancanti di interesse della propria organizzazione (anzi, nella filosofia Open Source, ogni utente può arricchire il progetto con propri contributi che saranno messi a disposizione di tutti)

### Flessibilità

Adottare un criterio di gradualità nella attuazione del CMDB vuol dire poter sviluppare il sistema attraverso fasi successive, in termini di struttura e di relazioni fra gli oggetti.

La necessità di gestire nuove tipologie di oggetti deve essere però risolta in modo autonomo dall'utilizzatore del sistema, senza ricorrere a costose modifiche del software.

La flessibilità di CMDBuild è stata perseguita come criterio principale, rendendo disponibile all'utilizzatore funzionalità per:

- aggiungere nuove "classi", cioè tipologie di oggetti (tabelle del DB)
- aggiungere / modificare gli "attributi" di una classe (colonne del DB)
- aggiungere "tipologie di relazioni" fra "classi" (tabelle di relazione n:m del DB)
- definire ruoli e autorizzazioni legati alle diverse categorie di "classi"

Ovviamente l'interfaccia per apportare modifiche strutturali al sistema sarà disponibile solamente agli utenti abilitati al ruolo di amministratore.

### Object Oriented

L'utilizzatore di un CMDB ragiona in termini di oggetti perché tali sono gli elementi che deve descrivere e gestire nel lavoro di tutti i giorni.

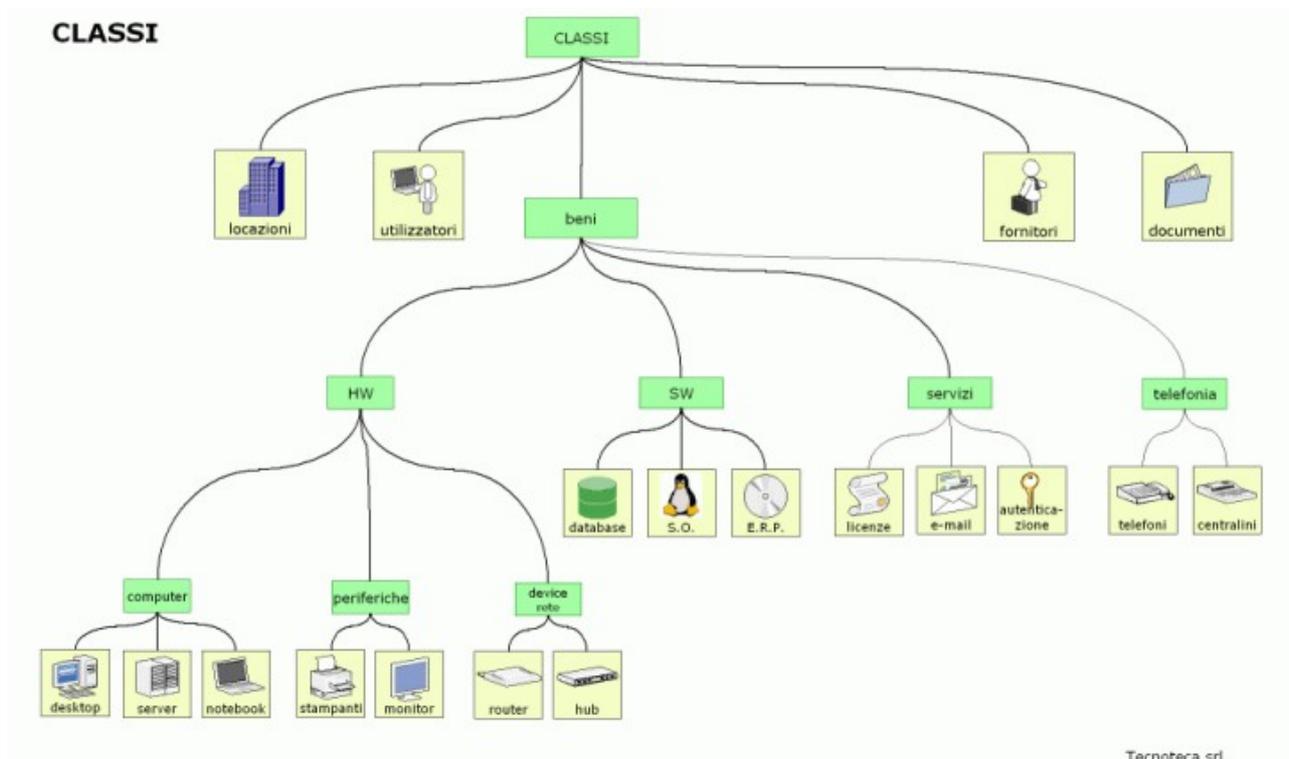
Computer, periferiche, applicazioni software, fornitori, licenze, servizi, documenti sono tutti oggetti da gestire in termini di caratteristiche, variazioni nel tempo e relazioni reciproche.

La soluzione naturale per mantenere l'applicazione coerente con le necessità dell'utilizzatore era quindi quella di ragionare in termini di oggetti anche nella progettazione e realizzazione di CMDBuild.

CMDBuild è realizzato con linguaggio Java, linguaggio di programmazione object oriented "puro", in cui ogni entità è vista come un oggetto e deriva da una classe principale.

CMDBuild utilizza pesantemente la caratteristica object oriented offerta dal database PostgreSQL, tramite una struttura gerarchica di classi dove ogni classe eredita gli attributi della superclasse estendendoli con propri attributi aggiuntivi e consentendo così di differenziare classi simili tra loro

(ad esempio dalla classe “cespiti” può essere derivata la sottoclasse “elaboratori” e da questa le ulteriori sottoclassi “server”, “desktop”, “notebook”, ecc).



## Specializzazione

CMDBuild fornisce di base un sistema “general purpose” finalizzato alla creazione autonoma delle basi di dati strutturate e relazionate con cui “modellare” gli oggetti IT di interesse.

Nell’ambito di tale sistema di base possono rendersi necessarie funzionalità “custom” specifiche per alcune tipologie di oggetti particolari, che potranno essere di interesse comune per tutti gli utenti o di interesse particolare per una organizzazione.

Un esempio semplice del primo tipo è la classe “asset” per la quale è richiesta la classificazione in categorie e sottocategorie (già disponibile nel sistema), un esempio più sofisticato è la classe “document” per la quale è utile disporre di funzionalità di indicizzazione “full text”, versioning, presentazione tramite interfaccia utente dedicata (futura estensione).

Eventuali esigenze di questo tipo sono previste e già predisposte nell’architettura del sistema tramite l’implementazione di classi java personalizzate assegnabili dall’amministratore del sistema come “gestore” specifico ad ogni tipologia di oggetto che lo dovesse richiedere.

## Interamente utilizzabile via web

CMDBuild è una applicazione web a tutti gli effetti, sia per le funzionalità di gestione e produzione di reportistica che per quelle di amministrazione e strutturazione del database.

L’utente del sistema deve disporre sul proprio elaboratore esclusivamente di un browser web di recente generazione.

La completa utilizzabilità web del sistema consente di supportare eventuali organizzazioni IT operanti in più sedi, consentendo l'accesso ai dati anche ad eventuali strutture esterne cui dovessero essere state affidati servizi di Help Desk in modalità di outsourcing.

## **Interoperabilità**

I dati archiviati nel CMDB devono essere accessibili, compatibilmente con le politiche di sicurezza definite, in lettura e scrittura da tutte le applicazioni interessate alla loro gestione, nell'ambito dei processi ITIL o dedicate ad esigenze amministrative.

Per garantirne la consultazione e modifica da applicazioni esterne è previsto lo sviluppo di un webservice SOAP nel quale saranno disponibili in modo controllato metodi di accesso esterno ai dati di CMDBuild.

Nell'ottica dell'interoperabilità con altri sistemi sono state implementate interfacce con applicazioni open source esterne specializzate in attività connesse a CMDBuild:

- Document Management (Alfresco)
- Automatic Inventory (OCS Inventory)
- Help Desk (DCL - Double Choco Latte)

## Funzionalità di CMDBuild

CMDBuild è costituito da due moduli principali:

- il Modulo Schema, dedicato alla definizione iniziale ed alle successive modifiche della struttura dati (classi e sottoclassi, attributi delle classi, tipologie di relazioni fra classi)
- il Modulo Gestione Dati, dedicato alla archiviazione ed aggiornamento nel sistema dei dati descrittivi e delle relazioni funzionali fra le diverse entità, alla gestione di documenti allegati, alla produzione di report e tabulati, nonché alla definizione e controllo dei processi per la gestione dei servizi informatici.

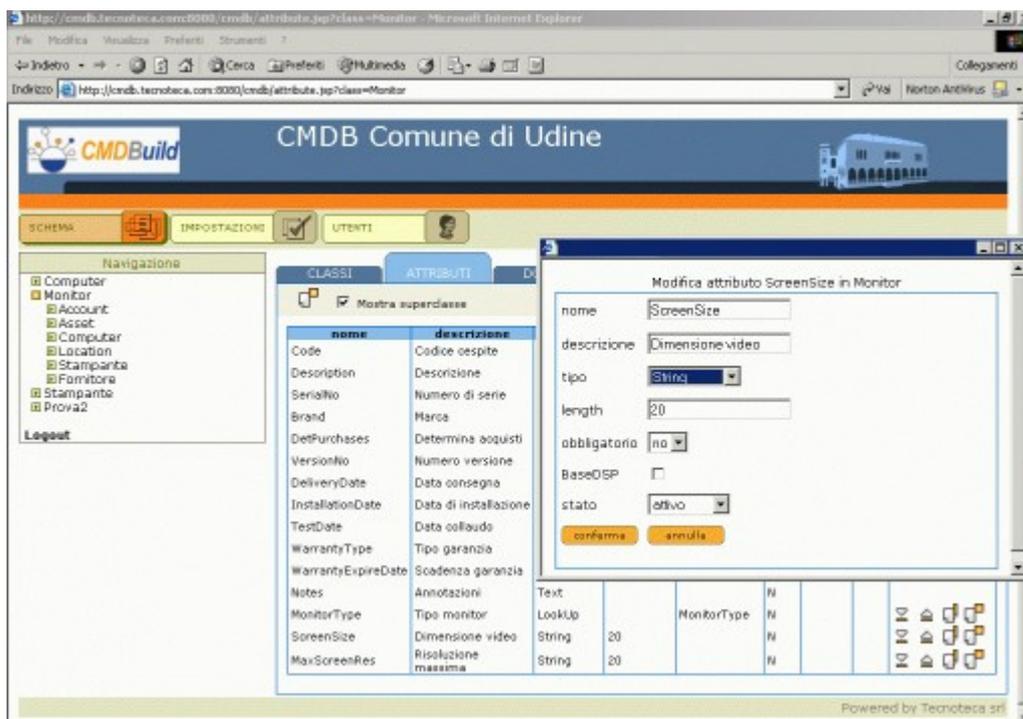
Per la gestione di documenti allegati ad ogni singola scheda di CMDBuild è stato utilizzato il repository del sistema di gestione documenti Alfresco.

Per la gestione di funzioni di Trouble Ticketing, previste da ITIL nell'ambito del processo di Incident Management, è stato personalizzato il prodotto Open Source DCL (Double Choco Latte) affinché possa interfacciarsi con il database di CMDBuild e produrre reportistica aggiuntiva.

Per l'inventariazione automatica degli asset e l'alimentazione del workflow con istanze del processo di "Change Management" è stato integrato il prodotto Open Source OCS Inventory.

### Principali funzionalità del modulo Schema

- creazione nel sistema di nuove "classi" (specificando l'eventuale superclasse da cui derivare parte dei propri attributi)
- creazione / modifica degli "attributi" di una classe (tipo booleano, intero, decimale, double, stringa, data, testo lungo, lookup a lista, riferimento)



- creazione di "tipologie di relazioni" fra "classi", con specifica e controllo cardinalità (uno a molti, molti a molti) e con flag specifico per la gestione Master-Detail delle classi interessate
- creazione di tipologie di voci tabellate ("lookup") per la gestione di attributi a valori fissi, compilabili con i valori di interesse di ogni organizzazione
- creazione di categorie e macrocategorie per la classificazione degli "Asset", compilabili con i valori di interesse di ogni organizzazione
- importazione report disegnati con editor esterni
- importazione workflow disegnati con editor esterni
- configurazione dei connettori con fonti dati esterne
- impostazione di parametri base di configurazione
- definizione di ruoli e autorizzazioni differenziabili per le diverse classi di oggetti

## Principali funzionalità del modulo Gestione Dati

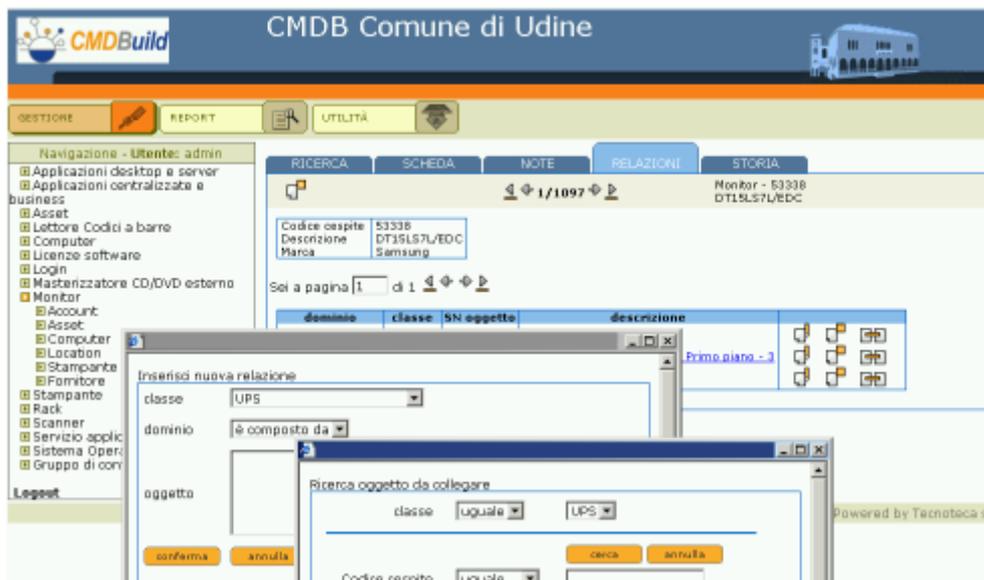
### Funzionalità base

- ricerche su una classe di oggetti specificando sia filtri sui dati e sul testo dei documenti collegati che criteri di relazione con altre classi
- creazione e modifica delle schede degli oggetti gestiti
- gestione di note collegate alle schede degli oggetti gestiti
- creazione e modifica delle relazioni fra gli oggetti gestiti
- caricamento di documenti allegati alle schede
- storicizzazione automatica delle modifiche ai dati

- consultazione della storia delle modifiche ai dati ed alle relazioni fra gli oggetti gestiti nel sistema

### **Funzionalità avanzate**

- gestione di processi associati alle classi trattate in CMDBuild
- wizard per l'utilizzo di funzionalità avanzate (copia di un CI a partire da un altro CI, modifica in blocco delle relazioni di un insieme di CI, ecc)
- produzione di report in formato PDF e CSV (costruzione guidata della query tramite navigazione fra le relazioni, definizione criteri di filtro con operatori logici, selezione attributi da stampare, scelta criteri di ordinamento, salvataggio report, esportazione query)
- esecuzione di report in formato PDF e CSV disegnati esternamente ed importati nel sistema
- modifiche cumulative sulle relazioni di un insieme di schede specificato
- import guidato di dati da file in formato CSV



## Motore di workflow

Per poter supportare le indicazioni metodologiche di ITIL il sistema CMDBuild deve non solamente supportare l'aggiornamento dell'inventario degli asset e delle relazioni funzionali fra di essi, ma anche consentire la definizione e controllo dei processi per la gestione dei servizi informatici.

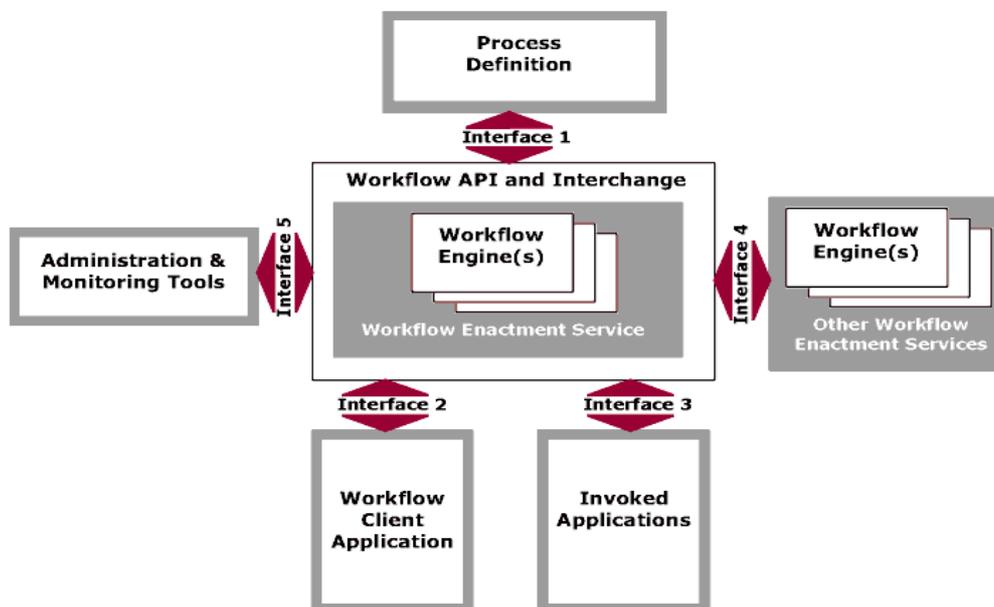
Un processo consiste di una o più attività, svolte da operatori e/o da applicazioni informatiche, ciascuna delle quali rappresenta un'azione da svolgere all'interno del processo, nel caso specifico relativamente alla gestione in qualità degli asset informatici.

Fra i processi fondamentali coperti da ITIL ci sono quelli del Service Management, comprendenti:

- Service Support: Incident Management, Problem Management, Change Management, Configuration Management, Release Management
- Service Level Management, Availability Management, Capacity Management, Continuity Management, Financial Management

Dati il numero elevato dei processi attivabili, le peculiarità organizzative dei singoli enti e le caratteristiche di estensibilità, flessibilità ed autonomia di gestione perseguite dal progetto, si è cercato di implementare in CMDBuild non una serie di processi rigidi e predefiniti, ma un sistema generico tramite il quale utenti evoluti possano disegnare ed attivare autonomamente i workflow di proprio interesse.

Segue lo schema di riferimento per la gestione dei processi secondo il modello standardizzato dal WfMC e adottato in CMDBuild.



Il sistema scelto in CMDBuild per la gestione del workflow utilizza i seguenti strumenti:

- XPDL come linguaggio di definizione (standardizzato dalla WfMC, WorkFlow Management Coalition sulla base del modello sotto riportato)
- il motore open source Enhydra Shark (<http://www.enhydra.org/workflow/shark/>), un framework estensibile che fornisce una implementazione completa e standard delle specifiche WfMC (<http://www.wfmc.org/>) e OMG, utilizzando al suo interno XPDL come linguaggio nativo
- l'editor visuale JPED (<http://jped.sourceforge.net/>) per il disegno del workflow e per la definizione dei meccanismi di integrazione con CMDBuild

Si rimanda allo specifico manuale di CMDBuild "Workflow Tutorial" per una descrizione dettagliata del sistema implementato.

## OCS Inventory

La raccolta manuale delle variazioni effettuate al numero ed alle caratteristiche dei beni hardware e software gestiti può introdurre problemi di ritardo o imprecisione nell'aggiornamento dei dati.

Per una gestione semiautomatica di tali aggiornamenti è stato individuato il prodotto Open Source OCS Inventory, in grado di:



- rendere disponibili programmi "agenti" da installare sui computer da controllare (con sistema operativo Windows, Linux o Mac OS)
- raccogliere le informazioni provenienti dagli agenti (relativamente a BIOS, processore, RAM, device di input, controller, periferiche, schede aggiuntive, impostazioni rete, sistema operativo, applicazioni software, ecc)
- archivarle in un proprio database

Il sistema di interfaccia fra CMDBuild e OCS Inventory è basata sui seguenti criteri:

- confronto periodico fra i dati archiviati in CMDBuild ed i dati rilevati da OCS Inventory, reso possibile dalla disponibilità di:

- una descrizione formale personalizzata del “mapping” fra i due insiemi di attributi (basata su file XML e XLST)
- API specifiche per l'interrogazione del database di CMDBuild
- alimentazione del sistema di workflow di CMDBuild con richieste di modifica automatiche (variante semplificata di un processo di Change Management) generate sulla base delle variazioni di configurazione rilevate

Il secondo criterio è ovviamente dipendente dai requisiti di controllo esplicito e di gestione dei ruoli e delle responsabilità richiesti da ITIL.

I criteri di “mapping” fra classi e relativi attributi presenti nei due sistemi sono personalizzabili (tramite una specifica applicazione visuale) per poter corrispondere alla struttura delle classi definita da ciascun ente in CMDBuild in funzione delle proprie necessità e della propria organizzazione interna.

Il sistema estremamente generico sopra descritto si presta per poter essere utilizzato in altri ambiti ove siano richieste funzionalità di aggiornamento periodico di CMDBuild da database esterni.

Tali ambiti possono comprendere sistemi di inventory diversi da CMDBuild, ma anche altre tipologie di basi dati, quali ad esempio archivi di dipendenti o per la gestione amministrativa dei cespiti.

Il meccanismo è stato ulteriormente esteso prevedendo la connessione con server di posta per ricevere aggiornamenti ai dati o avviare processi tramite email provenienti da uno specifico account.

Per ulteriori dettagli si veda lo specifico manuale di CMDBuild “External Connectors”.

## DCL (Double Choco Latte)

L'applicazione DCL (Double Choco Latte) comprende funzioni per il supporto del servizio di Call Center, dell'Help Desk di secondo livello e di attività programmate.

Pur disponendo CMDBuild già al suo interno di un sistema di workflow in grado di descrivere processi di gestione tramite un meccanismo “general purpose” basato sul linguaggio XPDL , può risultare che le funzionalità o l'interfaccia utente ottenibili dalla implementazione di un processo di “Help Desk” non siano ritenute sufficienti (vedi esempio descritto nel manuale “Workflow Tutorial”).



Il sistema di workflow interno a CMDBuild consente infatti la definizione di processi generici utili per le normali attività di gestione e garantisce la gestione integrata ed omogenea di classi e processi e delle relative correlazioni, ma può non comprendere le funzionalità specifiche disponibili in un sistema di Help Desk sofisticato.

Qualora si ritenesse preferibile trattare l'Help Desk esternamente a CMDBuild, è quindi disponibile l'interfaccia con il sistema open source DCL, che rende disponibili funzioni per:

- apertura nuovo ticket con classificazione, descrizione, caricamento file allegati (vedi screenshot successivo)
- creazione / assegnazione / modifica / annullamento / aggiornamento degli ordini di lavoro relativi ai ticket non direttamente risolvibili, con notifica via mail alle persone interessate
- registrazione in un apposito archivio di “timecard” (foglio di lavoro) degli orari e delle operazioni svolte
- utilizzo tabelle account, personale tecnico, dipartimenti personale tecnico, canale di ricezione / severità / priorità / stato della segnalazione, prodotti / moduli oggetto di segnalazioni

Le personalizzazioni effettuate da Tecnoteca sul codice sorgente di DCL riguardano:

- associazione di un ticket ad un asset ricercabile nella corrispondente tabella del database di CMDBuild
- associazione di un ticket ad un account da una lista popolata accedendo alla corrispondente tabella del database di CMDBuild

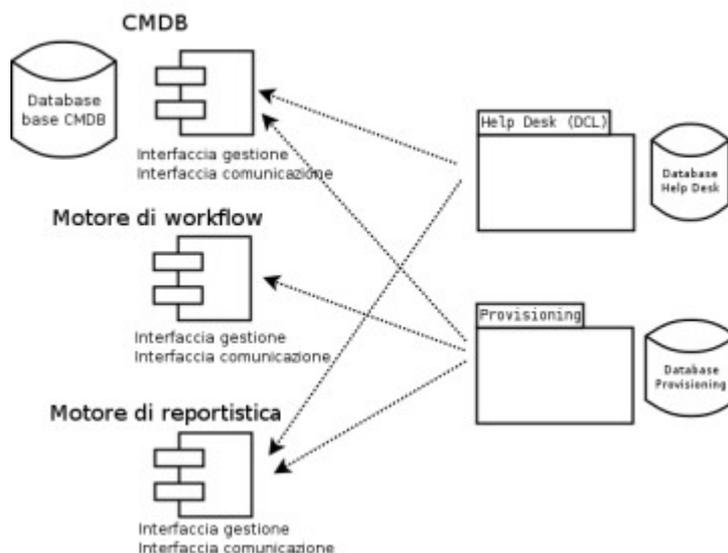
Le integrazioni effettuate consentono quindi di operare da DCL sugli asset e sui relativi assegnatari gestiti in CMDBuild, non sono al momento gestite ulteriori interazioni o integrazioni fra i due prodotti.

## Tecnologia di CMDBuild

Il sistema CMDBuild è interamente realizzato con componenti ed ambienti di sviluppo Open Source.

### Ambienti di sviluppo e produzione:

- linguaggio Java e ambiente di sviluppo Eclipse
- Apache Struts (implementazione design pattern MVC)
- motore reportistica JasperReports più editor visuale IReport
- motore workflow Shark Enhydra più editor visuale JPED
- sistema gestione documenti Alfresco
- sistema operativo Linux
- webserver Apache e Tomcat,
- database PostgreSQL



### Approfondimento: utilizzo avanzato di PostgreSQL

Gli obiettivi di base considerati nella progettazione di CMDBuild prevedevano:

- la gestione di un insieme dinamico di tabelle (corrispondenti alle diverse tipologie di CI) strutturate in una gerarchia di superclassi e sottoclassi per gestire in modo efficace eventuali insiemi di attributi comuni
- la gestione di un insieme dinamico di tabelle di relazione molti a molti (corrispondenti alle diverse tipologie di relazioni fra i CI, ad esempio "collegamento", "inclusione", "dipendenza funzionale", "composizione baseline", "utilizzo licenza", ecc)
- il completo tracciamento della storia delle modifiche di un CI nel tempo (sia degli attributi che delle relazioni)

Le tre esigenze sopra ricordate sono state risolte in modo naturale nell'ambito del database PostgreSQL, sfruttandone appieno la caratteristica object oriented nativa del database.

Attraverso il meccanismo di derivazione fra tabelle (keyword "inherits") è stato possibile, in modo assolutamente naturale:

- implementare la gerarchia delle superclassi e sottoclassi
- specializzare la generica classe di relazione "Map" in sottoclassi "Map\_xx" rappresentanti ciascuna una diversa tipologia di relazione definita nel sistema
- storicizzare le modifiche delle informazioni registrate nelle classi di dati e di relazioni, archiviandole tramite opportuni trigger in classi derivate (caratterizzate dal suffisso "\_history")



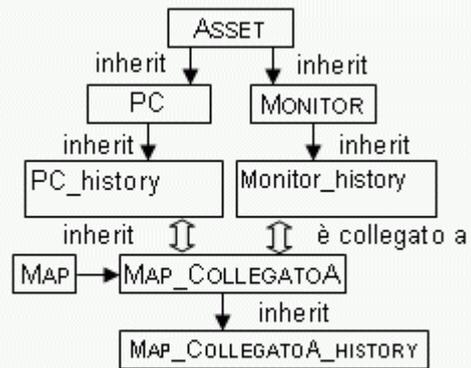
Esempio:

```

CREATE TABLE "Asset"
(
  "Id" numeric NOT NULL,
  "Description" varchar(250),
  "SerialNo" varchar(40),
  "VersionNo" varchar(32),
  "InstallationDate"
    timestamp,
  "WarrantyExpireDate"
    timestamp,
  "State" varchar(16),
  "StateDate" timestamp,
  CONSTRAINT asset_pkey
  PRIMARY KEY ("Id")
)

CREATE TABLE "Monitor"
(
  "MonitorType" varchar,
  "ScreenSize" varchar(16),
  "MaxScreenRes" varchar(16)
) INHERITS ("Asset")

CREATE TABLE
"Monitor_history"
(
  "HistoryId" numeric NOT
  NULL,
  "HistoryDate" timestamp
  NOT NULL DEFAULT now()
) INHERITS ("Monitor")
    
```



Idem per la catena "Map", "Map\_ContentutoIn", "Map\_ContentutoIn\_history"

## APPENDICE A: Glossario

### ALLEGATO

Per “allegato” si intende un qualunque file associabile ad una scheda dati inserita nel sistema.

Gli allegati contenenti testo (file PDF, Open Office, Microsoft Word, ecc) sono indicizzati in modalità full text consentendo la loro ricerca anche sulle singole parole contenute.

### ATTIVITA'

Per “attività” si intende uno dei passaggi che costituiscono il processo.

Una attività è caratterizzata da un nome, un esecutore, un tipo, eventuali attributi, eventuali metodi associati ad API di CMDBuild per poter essere eseguiti.

Per “istanza di attività” si intende una specifica attivazione di una attività, effettuata automaticamente dal sistema o manualmente da un operatore.

Vedi anche: Processo

### ATTRIBUTO

Il termine indica nel sistema CMDBuild la generica tipologia di informazione descrittiva di una determinata classe.

CMDBuild consente tramite il Modulo Schema di creare nuovi attributi in una classe o di modificarne alcune caratteristiche.

Nella classe “Fornitore” gli attributi sono ad esempio il nome, l'indirizzo, il numero di telefono, ecc.

Ogni attributo corrisponde nel Modulo di Gestione a campi di inserimento dati sulla apposita scheda di gestione della classe e a colonne della corrispondente tabella nel database.

Vedi anche: Classe, Superclasse, Tipo di attributo

### CI

Si definisce Configuration Item (Elemento della Configurazione) ogni elemento che concorre a fornire il servizio IT all'Utente, considerato ad un livello di dettaglio sufficiente per la sua gestione tecnica e patrimoniale.

Esempi di CI sono: server, workstation, programma applicativo, sistema operativo, stampante, ecc

Vedi anche: Configurazione

### CLASSE

Il termine rappresenta un tipo di dati complesso caratterizzato da un insieme di attributi che nel loro insieme descrivono quel tipo di dato.

Una classe modella una tipologia di oggetto da gestire nel CMDB, quale ad esempio un computer, una applicazione software, un servizio, un fornitore, ecc

CMDBuild consente all'Amministratore del Sistema, attraverso il Modulo Schema, di definire nuove classi e di cancellare o modificare la struttura di classi già definite.

Una classe è rappresentata a video da una apposita scheda di gestione dati e nel database da una tavola generata automaticamente al momento della definizione della classe.

Vedi anche: Scheda, Attributo

## **CONFIGURAZIONE**

Il processo di Gestione della Configurazione ha lo scopo di mantenere aggiornata e disponibile per gli altri processi la base di informazioni relativa agli oggetti informatici gestiti (CI), alle loro relazioni ed alla loro storia.

E' uno dei principali processi gestiti dal sistema ITIL.

Vedi anche: CI, ITIL

## **DATABASE**

Il termine indica un insieme di informazioni strutturato ed organizzato in archivi residenti sull'elaboratore server, nonché l'insieme dei programmi di utilità dedicati alla gestione dei tali informazioni per attività quali inizializzazione, allocazione degli spazi, ottimizzazione, backup, ecc.

CMDBuild si appoggia sul database PostgreSQL, il più potente, affidabile e completo database Open Source, di cui utilizza in particolare le sofisticate funzionalità e caratteristiche object oriented.

## **DOMINIO**

Un dominio rappresenta una tipologia di relazione fra una coppia di classi.

E' caratterizzato da un nome, dalle descrizioni della funzione diretta ed inversa, dai codici delle due classi e dalla cardinalità (numerosità degli elementi relazionabili) ammessa.

CMDBuild consente all'Amministratore del Sistema, attraverso il Modulo Schema, di definire nuovi domini e di cancellare o modificare la struttura di domini già definiti.

Vedi anche: Classe, Relazione

## **ITIL**

Sistema di "best practice" ormai affermatosi come "standard de facto", non proprietario, per la gestione dei servizi informatici secondo criteri orientati ai processi (Information Technology Infrastructure Library).

Fra i processi fondamentali coperti da ITIL ci sono quelli del Service Support, comprendenti l'Incident Management, il Problem Management, il Change Management, il Configuration Management ed il Release Management.

Per ogni processo considera la descrizione, i componenti di base, i criteri e gli strumenti consigliati per la misura della qualità del servizio, i ruoli e le responsabilità delle risorse coinvolte, i punti di integrazione con gli altri processi (per eliminare duplicazioni e inefficienze).

Vedi anche: Configurazione

## **LOOKUP**

Con il termine "LookUp" si indica una coppia di valori del tipo (Codice, Descrizione) impostabili dall'Amministratore del Sistema tramite il Modulo Schema.

Tali valori vengono utilizzati dall'applicazione per vincolare la scelta dell'utente, al momento della compilazione del relativo campo sulla scheda dati, ad uno dei valori preimpostati.

Il Modulo Schema consente la definizione di nuove tabelle di "LookUp" secondo le necessità dell'organizzazione.

## **PROCESSO**

Per “processo” si intende una sequenza di passaggi (“attività”) descritti nel sistema per svolgere in forma guidata e secondo regole prestabilite una determinata azione.

Per ogni processo saranno avviate in CMDBuild una serie di “istanze di processo”, una per ogni necessità di effettiva esecuzione dell'azione corrispondente, che avrà luogo su “asset” specifici e sarà svolta da utenti specifici.

Una “istanza di processo” viene attivata tramite avvio e conferma del primo passaggio previsto e termina alla esecuzione dell'attività finale prevista nella definizione.

Vedi anche: Attività

## **RELAZIONE**

Per “Relazione” si intende in CMDBuild un collegamento effettivo di due schede appartenenti a due classi, o in altri termini una istanza di un dato dominio.

Una relazione è quindi definita da una coppia di identificativi univoci delle due schede collegate e dall'identificativo del dominio utilizzato per il collegamento.

CMDBuild consente agli operatori del Sistema, attraverso il Modulo Gestione Dati, di definire nuove relazioni fra le schede archiviate nel database.

Vedi anche: Classe, Dominio

## **REPORT**

Il termine indica in CMDBuild una stampa (in formato PDF o CSV) riportante in forma analitica le informazioni estratte da una o più classi fra le quali sia definita una catena di domini.

I report possono essere generati e modificati dagli operatori di CMDBuild tramite una apposita funzione del Modulo di Gestione Dati e la relativa definizione viene memorizzata nel database per poter essere riutilizzata successivamente.

Vedi anche: Classe, Dominio, Database

## **SCHEDA**

Con il termine “Scheda” in CMDBuild si riferisce un elemento archiviato in una determinata classe.

Una scheda è caratterizzata da un insieme di valori assunti da ciascuno degli attributi definiti per la sua classe di appartenenza.

CMDBuild consente agli operatori del Sistema, attraverso il Modulo Gestione Dati, di archiviare nuove schede nel database e di aggiornare schede già archiviate.

Le informazioni di ogni scheda saranno memorizzate nel database alle opportune colonne di una riga della tavola generata per la classe di appartenenza della scheda.

Vedi anche: Classe, Attributo

## **SUPERCLASSE**

Una superclasse è una classe astratta utilizzabile per definire una sola volta attributi condivisi fra più classi.

Da tale classe astratta è poi possibile derivare classi reali che conterranno i dati effettivi e che comprenderanno sia gli attributi condivisi (specificati nella superclasse) che quelli specifici della sottoclasse.

Ad esempio è possibile definire la superclasse “Computer” con alcuni attributi base (RAM, HD, ecc) e le sottoclassi derivate “Desktop”, “Notebook”, “Server”, ciascuna delle quali con i soli attributi specifici.

Vedi anche: Classe, Attributo

### **TIPO DI ATTRIBUTO**

Ogni attributo definito per una determinata classe è caratterizzato da un “Tipo” che determina le caratteristiche delle informazioni contenute e la loro modalità di gestione.

Il tipo di attributo viene definito con il Modulo Schema e può essere modificato entro alcuni limiti dipendenti dalla tipologia dei dati già archiviati.

CMDBuild gestisce i seguenti tipi di attributo: “Boolean” (booleano, Si / No), “Date” (data), “Decimal” (decimale), “Double” (virgola mobile in doppia precisione), “Integer” (numero intero), “LookUp” (tabellato da lista configurabile in “Impostazioni” / “LookUp”), “Reference” (riferimento o foreign key), “String” (stringa), “Text” (testo lungo), “TimeStamp” (data e ora).

Vedi anche: Attributo

## APPENDICE B: Le Best Practice ITIL

Ai Sistemi Informativi si richiede da tempo un incremento di efficacia riducendone nel contempo il budget.

I Processi IT sono spesso non ben definiti come struttura, ruoli e responsabilità.

Da ciò un interesse sempre maggiore per la reingegnerizzazione dei processi secondo le migliori esperienze di funzionamento.

ITIL (Information Technology Infrastructure Library) fornisce un modello composto da linee guida e "best practice" per la gestione dei servizi informatici.

Diffusosi in tutto il mondo si è in poco tempo affermato quale "standard de facto", non proprietario, per la gestione dei servizi informatici, ispirando il nuovo BS 15000.

Le principali caratteristiche di ITIL:

- modello di riferimento per i Servizi IT orientato ai processi
- sviluppato per l'ICT del Governo inglese alla fine degli anni '80
- gradualmente arricchito in base alle esperienze degli utilizzatori
- approccio integrato alla gestione dei servizi (con definizione SLA)
- obiettivi: valorizzare l'IT in azienda migliorando i servizi e riducendone i costi

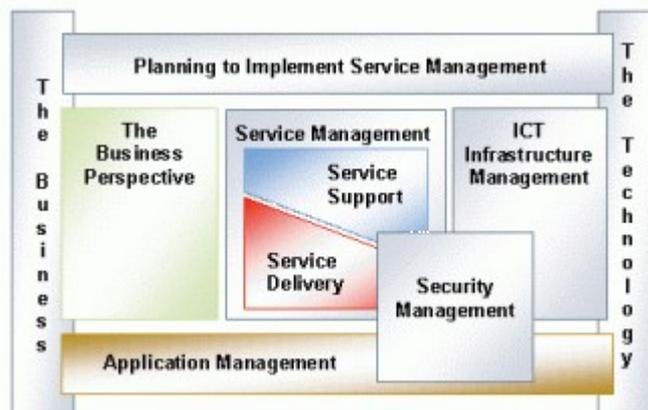


Per ogni processo considera la descrizione, i componenti di base, i criteri e gli strumenti consigliati per la misura della qualità del servizio, i ruoli e le responsabilità delle risorse coinvolte, i punti di integrazione con gli altri processi (per eliminare duplicazioni e inefficienze).

Fra i processi fondamentali coperti da ITIL ci sono quelli del Service Support, comprendenti:

- Incident Management
- Problem Management
- Change Management
- Configuration Management
- Release Management

CMDBuild si ispira alle "best practice" ITIL per la gestione del processo di Configuration Management, fornendo uno strumento informatico di supporto alle attività di strutturazione, aggiornamento e consultazione delle informazioni.



## APPENDICE C: Link utili

### SALPA 2005

Una prima presentazione del progetto CMDBuild al SALPA (Convegno nazionale 'Open Source e Pubblica Amministrazione') di Pisa

[www.salpa.pisa.it/salpa/22/02/36/SALPA\\_ATTACH\\_FILE220236.pdf](http://www.salpa.pisa.it/salpa/22/02/36/SALPA_ATTACH_FILE220236.pdf)

### Conferenza Annuale 2005 di itSMF Italia

Il Comune di Udine presenta il progetto di riorganizzazione dei processi informatici interni con il supporto di ITIL e del CMDBuild

[www.itsmf.it/download/GRUPPO01~ATTI\\_CONFERENZA\\_MILANO\\_2005/itSMF\\_ITIL\\_e\\_PA\\_Scaramuzzi.pdf](http://www.itsmf.it/download/GRUPPO01~ATTI_CONFERENZA_MILANO_2005/itSMF_ITIL_e_PA_Scaramuzzi.pdf)

### COMPA 2006

CMDBuild presentato al COMPA 2006 (Salone Europeo della Comunicazione Pubblica, dei Servizi al Cittadino e alle Imprese) di Bologna)

[http://www.cmdbuild.org/file/cmdbuild\\_compa\\_2006.pdf](http://www.cmdbuild.org/file/cmdbuild_compa_2006.pdf)

### Conferenza Annuale 2006 di itSMF Italia

Il Comune di Udine presenta l'avanzamento del progetto di riorganizzazione dei propri processi informatici interni con il supporto di ITIL e del CMDBuild

[http://www.cmdbuild.org/file/slide\\_itsmf\\_comuneudine.pdf](http://www.cmdbuild.org/file/slide_itsmf_comuneudine.pdf)

### Think Open 2007

Il Comune di Udine assieme ai partner del progetto presenta l'applicazione CMDBuild presso il proprio stand alla prima edizione della manifestazione Think Open tenutasi a Rovigo dal 26 al 28 settembre 2007.

Le slide delle presentazioni sono pubblicate in:

[http://www.cmdbuild.org/filesystem/Slide\\_ThinkOpen2007\\_Master\\_R03.pdf](http://www.cmdbuild.org/filesystem/Slide_ThinkOpen2007_Master_R03.pdf)

[http://www.cmdbuild.org/filesystem/Slide\\_ThinkOpen2007\\_Technical\\_R03.pdf](http://www.cmdbuild.org/filesystem/Slide_ThinkOpen2007_Technical_R03.pdf)

### Linux day 2007

Tecnoteca ha presentato l'applicazione CMDBuild al Linux Day 2007 nell'ambito del programma organizzato nella città di Udine, presso la locale Università, dal Gruppo Linux Udine (IGLU).

Le slide delle presentazioni sono pubblicate in:

[http://www.cmdbuild.org/filesystem/Slide\\_LinuxDay2007\\_Presentazione.pdf](http://www.cmdbuild.org/filesystem/Slide_LinuxDay2007_Presentazione.pdf)

[http://www.cmdbuild.org/filesystem/Slide\\_LinuxDay2007\\_ApprofondimentoTecnico.pdf](http://www.cmdbuild.org/filesystem/Slide_LinuxDay2007_ApprofondimentoTecnico.pdf)

### ALTRA PA

Il progetto è stato recensito fra le best practice di ALTRA PA, con descrizione dettagliata del progetto ed intervista al dirigente del Comune di Udine, suo committente, agli indirizzi:

<http://www.forumpa.it/archivio/3000/3800/3820/3827/cmdbuild-veloci.html> e

<http://www.forumpa.it/archivio/3000/3800/3810/3818/scaramuzzi-veloci.html>

### **Il sito ufficiale ITIL**

Sito di riferimento: [www.itil.co.uk/](http://www.itil.co.uk/)

### **ITIL IT Service Management Zone**

Sito di riferimento: [www.itil.org.uk/](http://www.itil.org.uk/)

### **L'Associazione itSMF Italia**

L'associazione "information technology Service Management Forum Italia" è un'organizzazione no-profit per la promozione e lo scambio di esperienze ed informazioni sulla gestione dei Servizi ICT e l'adozione delle migliori pratiche professionali secondo le indicazioni ITIL

Sito di riferimento: [www.itsmf.it/](http://www.itsmf.it/)

### **PostgreSQL (database)**

Sito di riferimento: [www.postgresql.org/](http://www.postgresql.org/)

### **Tomcat**

Sito di riferimento: [www.apache.org/](http://www.apache.org/)

### **Alfresco (repository documenti)**

Sito di riferimento: [www.alfresco.com/](http://www.alfresco.com/)

### **Enhydra Shark (motore workflow)**

Sito di riferimento: [www.enhydra.org/workflow/shark/](http://www.enhydra.org/workflow/shark/)

### **OCS Inventory (Automatic Inventory)**

Sito di riferimento: [www.ocsinventory-ng.org/](http://www.ocsinventory-ng.org/)