



CMDBuild

Open Source Configuration and Management Database Overview Document

**Versione 0.4.0
Maggio 2006**

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)



Cogitek S.r.l. (www.cogitek.it)



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

Sommario

Introduzione.....	4
Cos'è un CMDB.....	4
Perchè utilizzare un CMDB.....	4
Criteri di attuazione.....	5
Open Source.....	5
Le risposte di CMDBuild.....	6
Inventario dei beni informatici.....	6
Manutenzione del parco hardware.....	6
Gestione delle licenze software.....	6
Catalogo dei servizi.....	6
Tracciamento delle informazioni nel tempo.....	6
Gestione degli apparati di telefonia.....	7
Gestione della documentazione.....	7
Caratteristiche di CMDBuild.....	8
Flessibilità.....	8
Object Oriented.....	8
Interamente utilizzabile via web.....	9
Interoperabilità.....	9
Funzionalità di CMDBuild.....	10
Principali funzionalità del modulo Schema.....	10
Principali funzionalità del modulo Gestione Dati	11
Funzionalità base	11
Funzionalità avanzate.....	11
DCL (Double Choco Latte)	12
Call Center.....	12
Help Desk di secondo livello e manutenzione programmata.....	12
Reportistica.....	12
Integrazione di altre applicazioni	12
Tecnologia di CMDBuild.....	13
Ambienti di sviluppo e produzione:.....	13
Approfondimento: utilizzo avanzato di PostgreSQL.....	13
APPENDICE A: Le Best Practice ITIL.....	15
APPENDICE B: Link utili.....	16
SALPA	16
Conferenza Annuale 2005 di itSMF Italia	16
Il sito ufficiale ITIL	16
ITIL IT Service Management Zone	16
L'Associazione itSMF Italia	16

Introduzione

CMDBuild è una implementazione Open Source per la configurazione e gestione del database della configurazione (CMDB) degli oggetti 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.

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.

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 ?

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 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

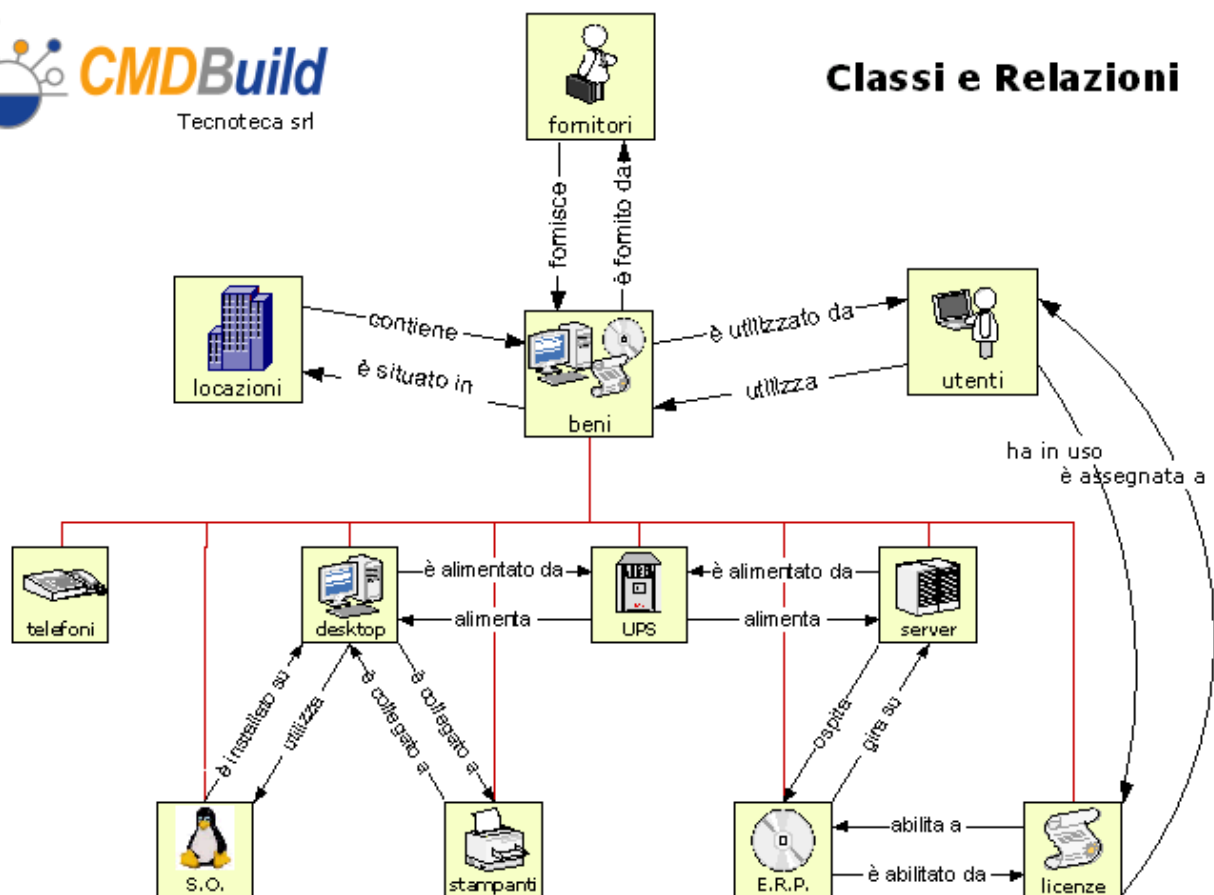
- quali utenti non hanno in dotazione un apparato telefonico ?
- ci sono prese telefoniche libere in una stanza dove è necessario spostare del personale ?

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 ?



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.

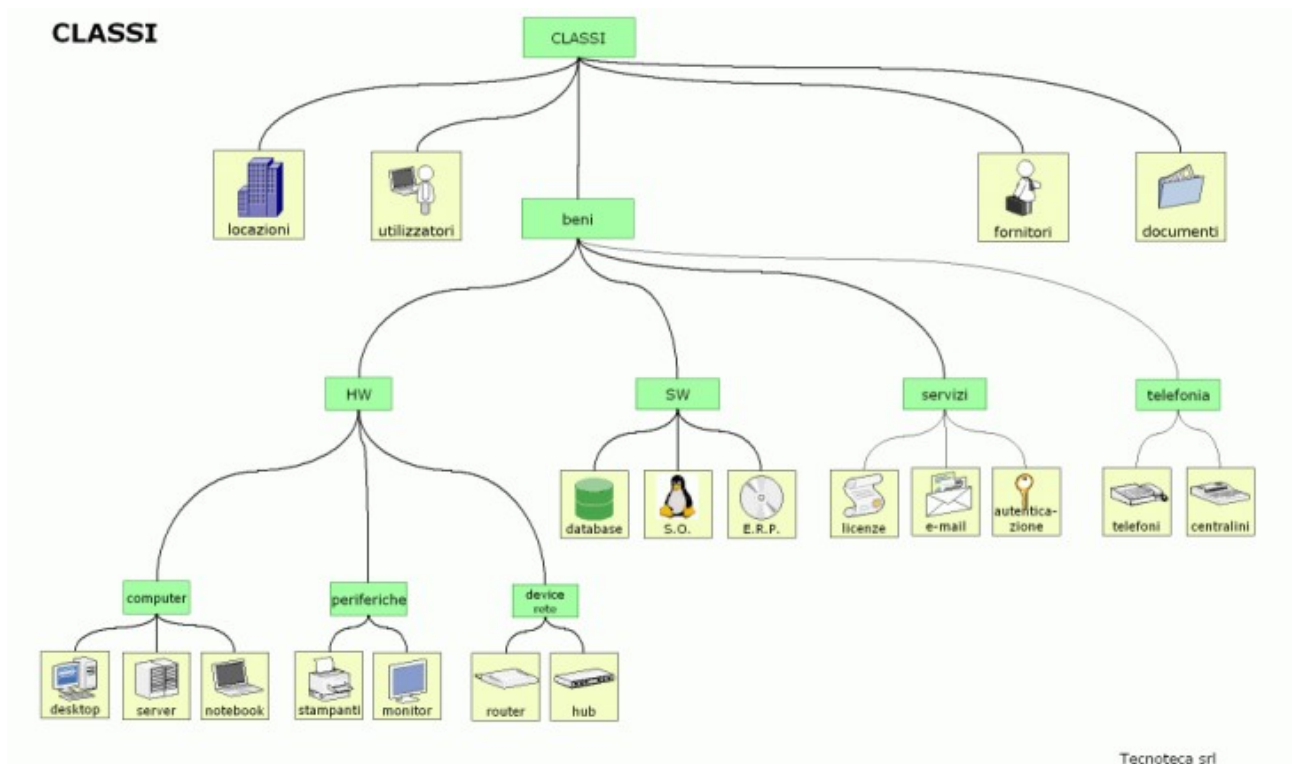
Computer, periferiche, applicazioni software, fornitori, licenze, servizi, documenti sono tutti oggetti da gestire in termini di caratteristiche, variazioni nel tempo e relazioni reciproche.

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).



Tecnoteca srl

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'utilizzatore 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.

Funzionalità di CMDBuild

CMDBuilds è costituito da due moduli principali:

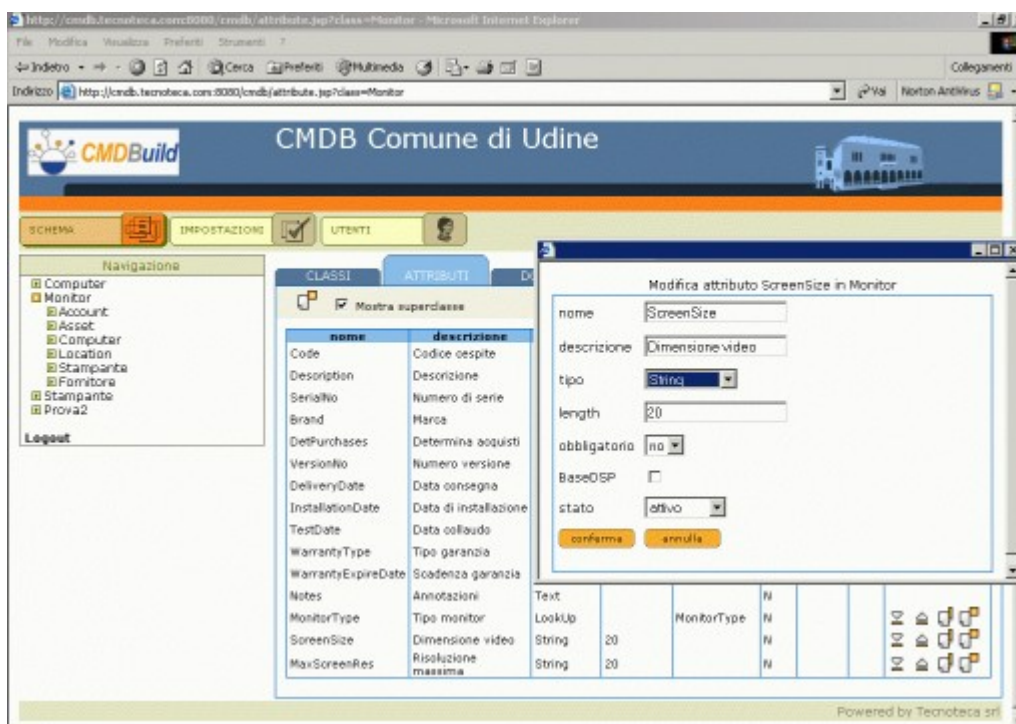
- il Modulo Schema, dedicato alla definizione iniziale ed alle successive modifiche della struttura dati (classi e sottoclassi, attributi delle classi, tipologie di relazioni fra classi)
- il Modulo Gestione Dati, dedicato alla archiviazione ed aggiornamento nel sistema dei dati descrittivi e delle relazioni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

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) per abilitarlo ad interfacciarsi con il database di CMDBuild e per produrre reportistica aggiuntiva.

E' inoltre prevista l'integrazione di CMDBuild con ulteriori applicazioni Open Source di supporto, quale ad esempio il sistema di Inventory ZCI e motori di workflow.

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 intero, float, stringa, data, boolean, testo lungo, lookup a lista)



- creazione di "tipologie di relazioni" fra "classi", con specifica e controllo cardinalità (uno a molti, molti a molti)
- creazione di tipologie di voci tabellate ("lookup") per la gestione di attributi a valori fissi, compatibili 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
- definizione di ruoli e autorizzazioni differenziabili per le diverse classi di oggetti
- esportazione dati
- cancellazioni logiche degli eventuali oggetti non più di interesse

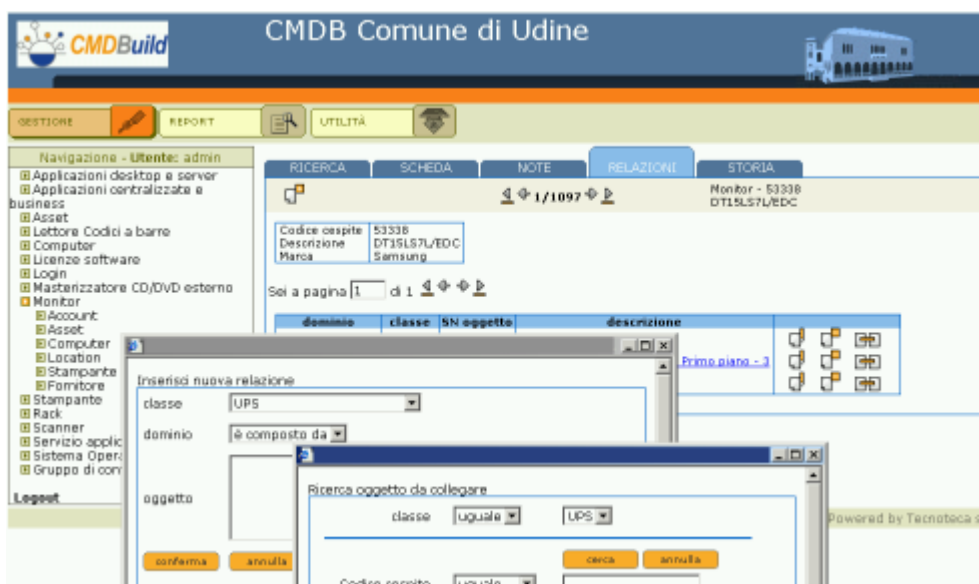
Principali funzionalità del modulo Gestione Dati

Funzionalità base

- ricerche su una classe di oggetti specificando sia filtri sui dati che criteri di relazione con altre classi
- creazione e modifica delle schede degli oggetti gestiti
- creazione e modifica delle relazioni fra gli oggetti gestiti
- storicizzazione automatica delle modifiche ai dati
- consultazione della storia delle modifiche ai dati ed alle relazioni fra gli oggetti gestiti nel sistema

Funzionalità avanzate

- 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, recupero dati da versioni storicizzate, salvataggio report, esportazione query)
- import guidato di dati da file in formato CSV



DCL (Double Choco Latte)

L'applicazione DCL, personalizzata per interfacciarsi con il database di CMDBuild, comprende funzioni per il supporto del servizio di Call Center, dell'Help Desk di secondo livello e della manutenzione programmata.

Nell'ambito delle attività di personalizzazione sono state inoltre realizzate funzioni di reportistica avanzata con utilizzo del motore Open Source Jasper Reports.



Call Center

- ricezione e registrazione nel sistema delle chiamate degli utenti (apertura ticket)
- classificazione della richiesta per successive attività statistiche
- fornitura diretta di una soluzione, ove possibile
- smistamento della richiesta, se necessario, alla struttura di assistenza tecnica specialistica (escalation), quale in particolare il servizio di Help Desk di 2° livello

Help Desk di secondo livello e manutenzione programmata

- creazione / assegnazione / modifica / annullamento / aggiornamento degli ordini di lavoro (workorder) relativi ai problemi trasmessi dal Call Center
- attribuzione dell'incarico ad un operatore
- esecuzione dell'attività
- registrazione in un apposito archivio di "timecard" (foglio di lavoro) degli orari e delle operazioni svolte

Reportistica

- tabulato ticket registrati con classificazione, stato corrente e informazioni di chiusura,
- stampa statistica con riepilogo mensile numero ticket aperti / risolti, aggregabili per canale di segnalazione, tipologia di attività, gravità, modalità di intervento, servizio di appartenenza dell'utente
- calcolo indici di valutazione del servizio (SLA) basati su indicatori predefiniti

Integrazione di altre applicazioni

ZMI è uno strumento Open Source per il recupero automatico via rete delle informazioni relative ai componenti hardware e software degli elaboratori e delle periferiche in uso. E' prevista l'integrazione in CMDBuild di una funzione di importazione dei dati rilevati e raccolti da ZMI, che sarà sviluppata in analogia con il wizard di importazione dati da file CSV già disponibile.



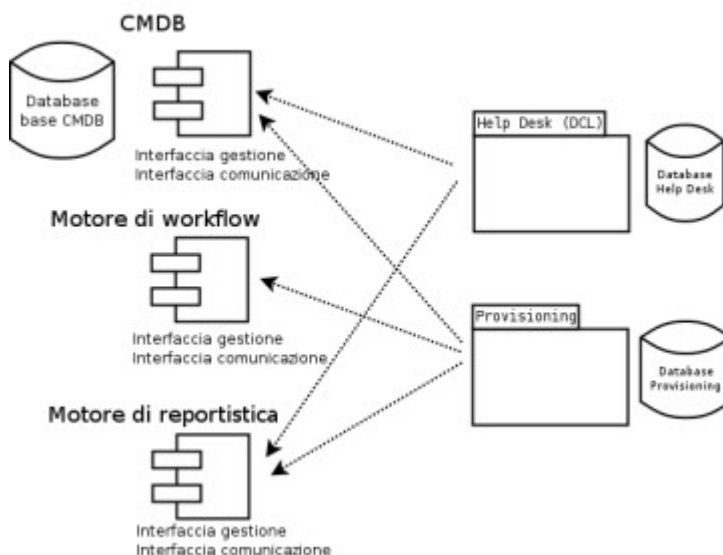
E' inoltre prevista l'integrazione in CMDBuild di funzionalità di workflow a partire da sistemi Open Source in corso di valutazione.

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)
- linguaggio PHP (modulo DCL integrato)
- motore reportistica JasperReports più editor visuale IReport
- motore workflow
- 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")



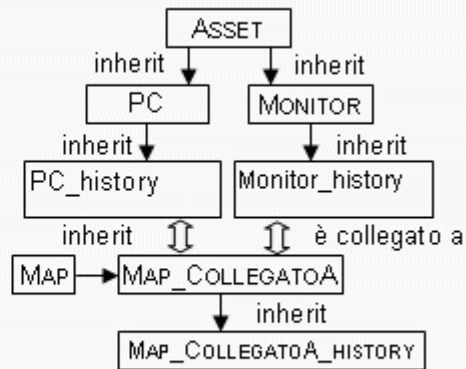
Esempio:

```

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: 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

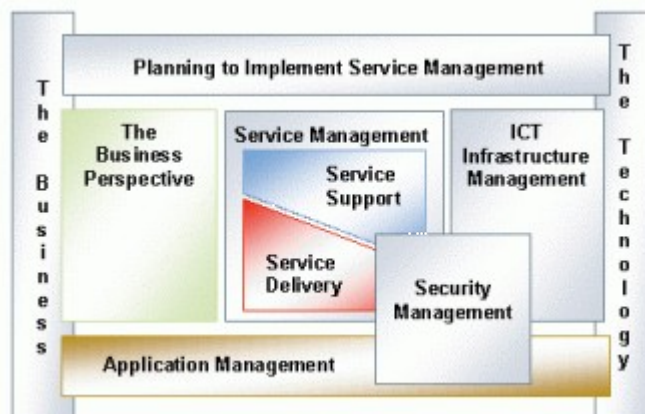


Per ogni processo considera la descrizione, i componenti di base, i criteri e gli strumenti consigliati per la misura della qualità del servizio, i ruoli e le responsabilità delle risorse coinvolte, i punti di integrazione con gli altri processi (per eliminare duplicazioni e inefficienze).

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 B: Link utili

SALPA

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

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

Il sito ufficiale ITIL

www.itil.co.uk/

ITIL IT Service Management Zone

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

www.itsmf.it/