

UNIVERSITA' DEGLI STUDI DI PISA
FACOLTA' DI SCIENZE MATEMATICHE FISICHE E NATURALI

Corso di Laurea in Scienze dell' Informazione

Tesi di Laurea

UN SUPPORTO AUTOMATICO PER LA RACCOLTA DEI REQUISITI

Candidati: Massimo Fontana

Relatore: Dott. Oreste Signore

Loredana Filippieri

Corelatore: Prof. Antonio Albano

Anno accademico 1984-85

Sommario	3
1.0 Introduzione.	2
1.1 Le metodologie di progettazione di Basi di Dati.	3
1.1.1 La metodologia DATAID.	8
1.2 Contenuti della tesi.	11
2.0 Strumenti automatici di documentazione.	14
2.1 Modalita' di interazione tra utilizzatore e strumento.	15
2.2 Schema concettuale della documentazione.	17
2.3 Alcuni strumenti di documentazione automatica.	18
3.0 Raccolta dei requisiti.	29
3.1 Aspetti trattati nella raccolta dei requisiti.	30
3.1.1 Aspetti strutturali.	30
3.1.1.1 La conoscenza concreta.	31
3.1.1.2 La conoscenza astratta.	34
3.1.1.3 La conoscenza procedurale.	34
3.1.2 Aspetti dinamici.	35
3.1.3 Aspetti quantitativi.	36
3.2 Definizione dei requisiti.	36
3.2.1 La definizione dei progetti e dei settori.	37
3.2.2 La definizione dei dati.	38
3.2.3 La definizione dei tipi.	41
3.2.4 La definizione delle operazioni.	41
3.2.5 La definizione delle attivita'.	42
3.3 Controlli.	45
3.3.1 Controlli di consistenza.	46
3.3.2 Controlli di completezza.	47
3.3.3 Controlli di adeguatezza.	48
4.0 Architettura del sistema.	49
4.1 Funzionalità di uno strumento automatico.	49
4.2 Scelta dell'ambiente hardware/software.	51
4.3 IL FOCUS: linguaggio della IV generazione.	52
4.3.1 L'Ambiente.	54
4.3.2 Definizione della base di dati.	55
4.3.3 Manipolazione dei dati.	58
4.3.4 Funzionalità del FOCUS.	69
5.0 Il sistema realizzato.	74
5.1 L'interfaccia utente-sistema.	74
5.1.1 Navigazione nel sistema.	78
5.2 Ambiente operativo.	83
5.3 Esempio di sessione.	86
5.4 La documentazione prodotta dal sistema.	106
5.5 I controlli realizzati.	150
6.0 Conclusioni.	156
Appendice A. Schema concettuale della metabase.	158
Appendice B. Schema logico e fisico della metabase.	161

B.1	Progetto	161
B.2	Settore	162
B.3	Classe	163
B.4	Attributo	164
B.5	Associazione	165
B.6	Tipo	166
B.7	Attivita'	167
B.8	Operazione	168
Appendice E. Controlli realizzati e loro gestione.		
C.1	Controlli sui progetti:	169
C.2	Controlli sui settori:	170
C.3	Controlli sulle classi:	172
C.4	Controlli sugli attributi:	175
C.5	Controlli sulle associazioni:	176
C.6	Controlli sui tipi:	180
C.7	Controlli sulle operazioni:	181
C.8	Controlli sulle attivita':	182
Bibliografia		184

SOMMARIO

Viene proposto un supporto automatico per la raccolta e l'analisi dei requisiti secondo quanto previsto dalle metodologie di progettazione di basi di dati.

Viene definita una base di dati per la memorizzazione dei requisiti sulla quale vengono realizzati dei controlli per garantire la corretta evoluzione.

Vengono realizzati rapporti di documentazione automatica.

1.0 INTRODUZIONE.

Il funzionamento di una organizzazione e' condizionato dalla accuratezza dei dati di cui dispone o dalla tempestivita' con cui possono essere utilizzati, sia per svolgere le proprie funzioni operative, sia per ricavare le informazioni necessarie ai processi decisionali e di controllo.

Ogni organizzazione pertanto, per la necessita' di produrle e di scambiare informazioni, fa uso di un sistema informativo, inteso come una combinazione di risorse umane e materiali impiegate per la raccolta, l'organizzazione, l'archiviazione e lo scambio di dati, sui quali viene svolta una attivita' di elaborazione per ricavare le informazioni necessarie al raggiungimento di determinati obiettivi.

Il progetto di un sistema informativo e' una attivita' molto complessa che inizia con l'analisi critica del modo in cui debbono essere soddisfatti i bisogni informativi della organizzazione. Noi ci poniamo nel contesto dei sistemi informativi basati sulla elaborazione automatica dei dati, indicando con sistema informatico il sottosistema informativo finalizzato a tale scopo.

Nel contesto dei sistemi informatici viene introdotto il concetto di base di dati:

Una Base di Dati e' un grande insieme integrato di dati, con una ridondanza minima e controllata, organizzato in modo tale da poter essere utilizzato, all'interno di una o piu' organizzazioni, da applicazioni diverse senza dipendere da alcuna di esse e rispettando alcune norme di confidenzialita'. [Albano, 85a]

1.1 LE METODOLOGIE DI PROGETTAZIONE DI BASI DI DATI.

Lo scopo del processo di progettazione e' quello di definire una base di dati che rappresenti accuratamente quella parte della realta' a cui siamo interessati, e tale da poter essere implementata in maniera efficiente su un sistema hardware/software esistente.

Disegnare una base di dati e' molto difficile, richiede una notevole quantita' di tempo e spesso non e' un processo strutturato.

La realizzazione di una base di dati e' una attivita' lunga e complessa che coinvolge non solo le funzioni di data processing, ma eventualmente l'intera organizzazione del sistema informativo.

La qualita' della base di dati risultante dipende molto dalle metodologie di disegno, dalle tecniche utilizzate nei

passi della metodologia, dalla validità dei requisiti informativi e dall'impiego di risorse dedicate. [Teorey,82]

Gli obiettivi della progettazione di una base di dati sono la realizzazione di una Struttura Dati (lo schema) che comprende tutte le informazioni atte a descrivere in modo integrato l'ambiente applicativo, e un insieme di transazioni (programmi) che agiscono su tale struttura svolgendo i compiti richiesti dalle varie applicazioni. [Ceri,83]

Riuscire a soddisfare in modo accettabile i requisiti di utenti diversi e' un compito assai complesso; peraltro appare evidente che gli errori di progettazione hanno una rilevanza estrema su tutta la vita del sistema informativo.

L'utilizzo di metodologie di progettazione, oltre a fornire un procedimento razionale e sistematico nell'identificazione dei problemi e nell'analisi completa delle esigenze applicative, fornisce utili strumenti per la documentazione e la verifica della qualità del progetto ed appare indispensabile nello sviluppo di sistemi informativi per realtà di per se' complesse.

In termini di basi di dati, una metodologia di disegno può essere pensata come un insieme di strumenti e di tecniche che possono essere applicate per sviluppare il progetto di una base di dati. [Masserman,80]

A causa della limitatezza degli strumenti a disposizione, e della scarsa conoscenza nel campo specifico delle basi di dati, le tecniche previste per la loro progettazione erano

fino a poco tempo fa organizzate in unica fase. La progettazione monofase e' generalmente informale, e cerca di soddisfare i fabbisogni informativi dell'utente utilizzando una qualche procedura di progettazione ad hoc.

Le problematiche riscontrate con questo approccio sono normalmente legate alla scelta di parametri fisici (problemi connessi alla memorizzazione dei dati, ecc.).

Ora tutti sono d'accordo sull'articolazione delle metodologie di progettazione in fasi di dati in piu' fasi.

In generale si possono individuare due momenti distinti: il primo di progettazione concettuale in cui i fabbisogni informativi degli utenti vengono analizzati al fine di giungere alla costruzione di un modello della base di dati (schema concettuale) che sia indipendente dallo specifico ambiente di realizzazione; il secondo di progettazione logico-fisica durante il quale lo schema concettuale viene tradotto nelle strutture logiche e fisiche del sistema di gestione utilizzato.

Piu' dettagliatamente [Tao,78] quattro fasi coprono tutti gli aspetti del processo di disegno. Esse sono:

- **raccolta e analisi dei requisiti:**

vengono raccolti e analizzati i fabbisogni informativi degli utenti. Lo scopo e' quello di definire il comportamento atteso del sistema da realizzare. Il risultato deve essere una specifica accurata e coerente dei requisiti, verificabile dagli utenti e sufficiente per impostare le fasi successive.

- **progettazione concettuale:**

in base alla specifica realizzata nella fase precedente si definisce il sistema da realizzare. Il risultato di questa fase e' detto "Progetto Concettuale". E' possibile esprimere il Progetto Concettuale attraverso una serie di documenti scritti in un linguaggio semi-formale, oppure utilizzando un linguaggio di tipo matematico oppure definendo una descrizione formale in un linguaggio di programmazione che abbia dei costrutti ad alto livello piu' vicini possibile all'ottica dell'utente. In questo ultimo caso e' possibile ottenere un prototipo del sistema che si vuole realizzare.

- **progettazione logica:**

riguarda la conversione del progetto concettuale nelle strutture dati ed operatori del Sistema di Gestione di Basi di Dati utilizzato.

- **progettazione fisica:**

riguarda tutte le attività che modificano o completano il progetto logico prendendo in considerazione aspetti del sistema relativi all'organizzazione fisica del dati.

L'organizzazione del processo di disegno in quattro fasi e' un punto di sostanziale accordo tra i ricercatori del settore, e un grande numero di metodi e strumenti sono stati proposti e sviluppati per definire e risolvere i vari aspetti del problema.

Le prime due fasi della metodologia non vengono influenzate dagli specifici sistemi di gestione di basi di dati (SGBD) o dagli ambienti software in cui le applicazioni sono disegnate.

La progettazione logica invece varia a seconda del modello dei dati del SGBD che si utilizza.

La progettazione fisica tiene conto delle caratteristiche interne del SGBD selezionato e dell'hardware.

In generale le metodologie di progettazione richiedono che la pianificazione delle applicazioni debba essere realizzata prima dell'inizio del processo del disegno di Basi di Dati. Aspetti molto importanti, come l'analisi della organizzazione e i suoi flussi informativi, e l'analisi di costi-benefici delle applicazioni su Basi di Dati non sono invece contemplati nei passi delle metodologie.

Molti passi delle fasi del processo di disegno possono essere assistiti con un opportuno utilizzo di tecniche di modellazione computerizzate ed altri strumenti, come un'analisi dei requisiti automatica. Questi strumenti spaziano tra meccanismi di documentazione e sofisticate metodologie di disegno.

1.1.1 La metodologia DATAID.

DATAID [Ceri,83] è una metodologia per la progettazione di basi di dati che copre tutte le fasi di progetto, dalla raccolta e l'analisi dei fabbisogni informativi alla specifica del corrispondente schema fisico in termini delle strutture

caratteristiche del sistema di gestione di basi di dati scelto.

Questa metodologia riflette la decomposizione del disegno in quattro fasi definite in [Yao,78]: raccolta e analisi dei requisiti, disegno concettuale (disegno di viste e loro integrazione), disegno logico, disegno fisico.

Nella fase di raccolta e analisi dei requisiti viene esaminata una descrizione in linguaggio naturale delle applicazioni sulla base di dati per ciascun settore aziendale relativamente ai dati, alle operazioni su di essi ed agli eventi che regolano la esecuzione di tali operazioni, in una forma ancora indipendente da metodi di rappresentazione formali; tale descrizione viene successivamente filtrata e classificata.

Il risultato di questa fase e' la realizzazione di tre glossari: glossario dati, glossario operazioni e glossario eventi.

La fase successiva (disegno concettuale) sfrutta i glossari realizzati nella prima fase per definire degli schemi strutturati per tutti i settori aziendali; viene infine realizzato un unico schema, prodotto dalla integrazione di tutti gli schemi definiti. Durante il processo di integrazione si verifica la consistenza delle diverse rappresentazioni e si risolvono eventuali conflitti.

Queste due fasi della metodologia non vengono influenzate da specifici sistemi di gestione di base di dati o dagli ambienti software in cui le applicazioni sono disegnate.

Il disegno logico e il disegno fisico sono definiti in funzione del sistema di gestione di basi di dati scelto.

L'applicazione della metodologia DATAID ha termine con il disegno di un sistema che soddisfi i bisogni degli utenti.

I principali vantaggi di DATAID si possono riassumere in questi termini:

- generalità e completezza;
- l'utilizzo di un modello di facile comprensione per l'utente; il modello è una estensione del modello Entità-Relazione [Chen,76] a cui vengono aggiunti meccanismi di estrazione.
- la specifica contemporanea dei requisiti nella forma di dati, transazioni ed eventi; le definizioni sono successivamente raffinate e vengono realizzati controlli di mutua consistenza a differenti livelli della metodologia.
- lo sviluppo dei disegni logico e fisico secondo criteri rivolti alla scelta di una buona rappresentazione tra diverse alternative che sono ricavate dallo schema concettuale.

La grande quantità di documenti prodotti ha come conseguenza molta difficoltà ed elevato consumo di tempo per gestire la documentazione e mantenerne la consistenza.

1.2 CONTENUTI DELLA TESI.

Il problema relativo al disegno di una base di dati può essere riassunto in un numero di problemi che sorgono nel ciclo di vita di un'applicazione [Teorey, #2]

1. cosa sono i requisiti di utente e come possono essere espressi;
2. come questi requisiti possono essere tradotti in strutture della base di dati;
3. come queste strutture si adattano ad una nuova specifica dei requisiti o ad un loro cambiamento.

Appare evidente l'importanza della raccolta dei requisiti nel ciclo di vita del processo di progettazione di una base di dati.

Lo scopo del lavoro svolto nell'ambito di questa tesi e' quello di fornire un supporto automatico per la raccolta dei requisiti.

Il sistema realizzato puo' essere utilizzato per trattare alcuni aspetti della progettazione concettuale, ed e' possibile specificare con esso alcune informazioni utili alla definizione dello schema fisico.

Cio' che e' stato realizzato puo' essere quindi visto come un prodotto di aiuto nella progettazione di basi di dati.

Una caratteristica del prodotto e' che opera su "mainframe", ma con piccole modifiche e' funzionante anche su "personal computer".

Per sviluppare il supporto automatico realizzato e' stato scelto un us linguaggio della IV generazione, il FOCUS. L'importanza e la difficulta' della comunicativa con gli utenti finali non deve essere sottovalutata [Teoney,92]

Nel capitolo 2 vengono descritti alcuni strumenti di documentazione automatica.

Nel capitolo 3 viene descritto cio' che e' di interesse nell'ambito del lavoro svolto ed il modo in cui tale realta' e' stata modellata.

Nel capitolo 4 viene descritto l'ambiente in cui e' stato sviluppato il supporto automatico realizzato. Viene presen-

tato il linguaggio utilizzato per la creazione di tale supporto.

Nel capitolo 5 viene descritto il sistema realizzato.

Nel capitolo 6 vengono descritti gli aspetti rilevanti della realizzazione.

La tesi prevede diverse appendici.

In appendice A viene presentato lo schema concettuale della metabase utilizzata per la raccolta dei requisiti.

In appendice B e' presentato lo schema logico-fisico della metabase.

In appendice C vengono descritti i controlli realizzati e la loro gestione.

2.6 STRUMENTI AUTOMATICI DI DOCUMENTAZIONE.

Ad ogni fase del ciclo di vita di un sistema informatico e' utile mantenere della documentazione su cio' che e' stato fatto.

La realizzazione e il mantenimento della documentazione sono necessarie sia ai progettisti sia agli utenti della base dati.

Nel caso di realizzazione di sistemi di grandi dimensioni la documentazione risulta indispensabile sia ai progettisti interessati alla stessa fase di progetto sia tra gruppi di progettisti coinvolti in fasi differenti.

Per quanto riguarda gli utenti, la documentazione fornisce loro la possibilita' di controllare la rispondenza del progetto ai requisiti da loro stabiliti, e di verificare lo stato di avanzamento del progetto.

L'utilizzo del calcolatore per gestire la documentazione presenta dei vantaggi rispetto alla documentazione manuale relativamente a:

1. la qualita' della documentazione;
2. la completezza della descrizione;
3. il controllo di ridondanza;

4. la semplicità di manutenzione.

Le caratteristiche principali di uno strumento di documentazione sono:

1. modalità di interazione tra utilizzatore e strumento;
2. schema concettuale della documentazione cui si deve fare riferimento per poter utilizzare lo strumento.

2.1 MODALITÀ DI INTERAZIONE TRA UTILIZZATORE E STRUMENTO.

Le modalità di interazione tra utilizzatore e strumento comprendono il linguaggio e i linguaggi con cui l'utilizzatore può creare o aggiornare la propria documentazione ed i tipi di rapporti che lo strumento può produrre. Il primo aspetto riguarda le modalità mediante le quali l'utilizzatore descrive le proprie specifiche allo strumento, il secondo è relativo alle elaborazioni che lo strumento è in grado di fare sulle specifiche realizzate.

La descrizione delle specifiche può essere effettuata utilizzando [Battini,84] :

1. un linguaggio pseudo-naturale;

2. un insieme di comandi parametrici;

3. un insieme di menu' e moduli;

4. un'interfaccia grafica.

L'utilizzazione di un linguaggio pseudo-naturale presenta il vantaggio di descrivere la specifica con un formalismo simile ai linguaggi utilizzati nella vita comune; lo svantaggio riscontrato e' la verosita' delle dichiarazioni.

Utilizzando un linguaggio basato su comandi parametrici ogni tipo di specifica e' espresso in modo sintetico allo scopo di rendere compatta la descrizione.

Se la descrizione delle specifiche avviene tramite menu e moduli l'utente e' guidato dal sistema a scegliere le operazioni da effettuare, i parametri da specificare ed i loro valori.

Le interfacce grafiche possono considerarsi come una alternativa al menu, volta a rendere piu' gradevole l'interazione con il sistema.

2.2 SCHEMA CONCETTUALE DELLA DOCUMENTAZIONE.

Lo schema concettuale della documentazione e' l'insieme dei tipi di concetti e dei tipi di legami logici tra tali concetti rappresentabili per mezzo dello strumento di documentazione; fissato lo schema, il compito del progettista e' quello di specificare gli aspetti del progetto che vuole documentare per mezzo dei tipi e disposizione dello schema [Batini,84] .

Puo' succedere che il modello di progetto non coincida con il modello di documentazione; ad esempio nel caso del lavoro svolto in questa tesi, puo' succedere che la raccolta dei requisiti sia stata realizzata prendendo come modello di progetto il Galileo/3 [Mavazzo,85], e di dover quindi individuare una corrispondenza tra coppie di categorie del modello di progetto e del modello concettuale dello strumento realizzato(vedi appendice 3).

In genere questa corrispondenza tende a degradare le proprieta' semantiche definite per il modello di progetto. Esistono strumenti di documentazione che permettono di adattare il modello di documentazione al modello di progetto; i risultati sono soddisfacenti dal punto di vista sintattico, ma risultano inadeguati dal punto di vista semantico.

2.3 ALCUNI STRUMENTI DI DOCUMENTAZIONE AUTOMATICA.

In questo paragrafo prenderemo brevemente in considerazione alcuni strumenti automatici esistenti con lo scopo di individuarne alcune caratteristiche; in particolare verranno confrontate le funzionalità di alcuni di essi.

- GDOC

GDOC [Batini,84] è uno strumento per la progettazione di basi di dati, funzionante su personal computer, realizzato utilizzando il sistema relazionale DBASE II. Questo strumento aiuta il progettista di una applicazione base dati durante la progettazione concettuale e, in parte, anche nella fase di sviluppo della applicazione, dal momento in cui inizia la fase di produzione del programma.

GDOC è composto da vari sottosistemi:

1. un sottosistema di aggiornamento della documentazione;
2. un sottosistema di produzione rapporti;
3. un sottosistema grafico;

4. un sottosistema per la generazione automatica delle transazioni.

Nel sottosistema di aggiornamento e' possibile selezionare il tipo di oggetto su cui operare interagendo con un menu' grafico. Ad ogni sessione viene mostrato l'intero metascema; una volta selezionato l'oggetto, e' possibile specificarne gli attributi riempiendo un modulo. Di seguito, e' possibile ritornare al menu' principale, oppure navigare nella parte del metascema adiacente al concetto appena considerato.

Attualmente sono implementate due tipi di verifiche: non e' possibile creare un altro oggetto con lo stesso identificatore e viene fatta una ricerca sugli attributi duplicati.

Per quanto concerne la documentazione, e' possibile stampare tre tipi di documentazioni:

1. Documentazione sintetica, cioe' dare una lista per ogni oggetto specificato nello schema concettuale della documentazione.
2. Documentazione estesa su ogni oggetto, cioe' per ogni oggetto dare l'insieme degli oggetti a lui logicamente correlati secondo lo schema concettuale della documentazione.

3. documentazione globale, intesa come l'unione delle documentazioni precedenti.

Per quanto riguarda il sottosistema grafico, l'utente puo' costruire sul video l'immagine diagrammatica dello schema, mediante primitive.

In fine, ogni volta che una sessione di documentazione viene chiusa e come risultato viene prodotto uno schema concettuale parziale, e' possibile accedere a tutte le entita' e relazioni dichiarate nello schema utilizzando operazioni di creazione, cancellazione, aggiornamento di istanze (viene stampato un modulo in cui compaiono tutti gli attributi dell'oggetto selezionato).

GDCC presenta le seguenti funzionalita':

1. il modello concettuale di GDCC e' un modello tipico per proiezione concettuale, il modello Entita'-Relazione [Chen,76]
2. in tutte le fasi di utilizzazione dello strumento viene adottato il simbolismo grafico del modello Entita'-Relazione, allo scopo di sfruttare la sua espressivita'.

3. Lo strumento permetta al progettista di operare in modo interattivo.

4. qualunque utente che conosca il modello Entita'-Relazione puo' progettare e produrre una applicazione.

PSL/PSA

Problem Statement Language (PSL) e Problem Statement Analyzer (PSA) [Lenzerini,84] vengono utilizzati per descrivere le specifiche di un sistema informativo.

Il PSL e' un linguaggio strutturato; la descrizione di un sistema in PSL consiste nella combinazione di frasi formali nel linguaggio strutturato e di testi in linguaggio naturale.

Descrivere un sistema informativo in PSL significa definire un insieme di oggetti e di relazioni tra essi. Gli oggetti e le relazioni sono tipizzati, nel senso che ognuno di essi appartiene ad una categoria con specifiche proprieta'.

La descrizione di un sistema in PSL e' divisa in sezioni, ciascuna delle quali e' dedicata alla definizione di un oggetto, delle sue proprieta' e delle relazioni con altri oggetti.

PSA e' un modulo software che analizza le frasi espresse in PSL, interpretandole e costruendo una base di dati

destinata a contenere la descrizione del sistema; e' inoltre in grado di effettuare operazioni di modifica della propria base di dati, e di produrre diversi tipi di rapporti di documentazione.

La descrizione data in ESI viene sottoposta ad una analisi di correttezza da parte del PSA che comprende, oltre al controllo di eventuali errori sintattici, la rilevazione di possibili incoerenze tra diverse frasi (ad esempio, assegnazione di tipi differenti allo stesso oggetto, legami di tipo illecito tra oggetti).

Il PSA puo' effettuare ulteriori analisi nella fase di sviluppo del sistema per verificare lo stato di avanzamento del sistema (ad esempio, controllo che ad ogni oggetto sia stato assegnato un tipo, oppure che, per oggetti di tipo specifico, siano stati definiti prefissati tipi di relazioni).

Il PSA e' inoltre in grado di produrre diversi tipi di rapporti di documentazione:

1. rapporti di sintesi sul sistema descritto;
2. rapporti di analisi sui singoli oggetti componenti il sistema;
3. rapporti grafici.

Confrontiamo gli strumenti descritti relativamente alle parti comuni:

- La filosofia di interazione fra utente e sistema e' sostanzialmente batch, nel senso che una sessione tipica di aggiornamento della documentazione si svolge descrivendo oggetti in ESL, che vengono periodicamente ad aggiornare la descrizione sulla base di dati; invece il GDOC permette al progettista di formalizzare interattivamente le descrizioni concettuali.
- L'interazione avviene secondo modalita' diverse: nel PSL/PSA attraverso frasi in linguaggio naturale ristretto mentre in GDOC vengono utilizzati dei menu' grafici.
- il PSL/PSA funziona su "Mainframe", mentre il GDOC e' stato sviluppato su "Personal".
- Le verifiche di consistenza risultano buone nel PSL/PSA, meno buone in GDOC.
- Per quanto riguarda la documentazione, il PSL/PSA fornisce vari tipi di rapporti, anche parametrici, mentre il GDOC realizza solo due tipi di rapporti.
- ambedue i sistemi sono forniti di un supporto grafico molto ricco.

GALILEO/R

Nell'ambito del progetto Galileo [Allano, 85c] e' stato studiato un insieme di strumenti integrati di ausilio al progettista in un ambiente interattivo. Tali strumenti sono stati sviluppati per risolvere le problematiche relative alle prime due fasi del processo di progettazione di basi di dati.

Per quanto riguarda la raccolta e l'analisi dei requisiti viene utilizzato un linguaggio, il Galileo/S, di stretta derivazione dal Galileo (linguaggio utilizzato nella fase di progettazione concettuale).

Il Galileo/R e' stato sviluppato tenendo conto che:

1. deve permettere di definire tutte le informazioni necessarie ai passi successivi della progettazione;
2. deve fornire diverse forme di rappresentazione delle specifiche;

Gli strumenti sviluppati per supportare il Galileo/R si basano sulle seguenti ipotesi:

1. descrivere i requisiti a diversi gradi di dettaglio;

2. utilizzare meccanismi di astrazione espressivi;
3. una immissione graduale delle definizioni;
4. una facile modalità di interazione;
5. una facile consultazione delle definizioni.

Il sistema di supporto del linguaggio prevede le seguenti componenti:

- un editore grafico orientato a forme strutturate;
- una metabase di dati per memorizzare le definizioni dei requisiti;
- il gestore della metabase di dati;
- il generatore di documentazione;
- l'analizzatore della specifica dei requisiti;
- il gestore delle interazioni con l'utente.

L'utente interagisce con il sistema per mezzo di un video grafico e di un dispositivo di selezione; ha la possibilità di creare, cancellare, visionare oggetti del si-

stema; inoltre, prima di modificare lo stato della metabase, può editare, stampare, validare le definizioni dei requisiti.

Gli oggetti del sistema sono: progetti, ambienti, classi, sottoclassi, operazioni, attività e tipi definiti. Gli oggetti sono in relazione tra loro. Ad esempio ogni ambiente è definito in un progetto; le operazioni, le attività e le classi sono in relazione tra loro e con l'ambiente in cui sono stati definiti. Il sistema permette all'utente di accedere agli oggetti associati a quello corrente sia in visione che in modifica con la stessa modalità utilizzata in [Ratini,84]

Lo stato del sistema è fissato da un progetto corrente, un ambiente corrente e un oggetto corrente (diverso dai precedenti).

Sono previsti degli help-online per aiutare il progettista nell'utilizzo del sistema.

I dati gestiti dal sistema sono elementi di liste; è possibile prendere visione del contenuto di una lista utilizzando una finestra sullo schermo. Sono previsti i seguenti tipi di liste:

- liste di informazione Sistema, contenenti informazioni sullo stato degli oggetti definiti dal sistema;

- liste di Annotazioni, contenenti annotazioni sugli oggetti definiti;
- liste di Validazione, contenenti gli errori riscontrati sulla specifica dei requisiti;
- liste di Specifica Requisiti, contenenti le specifiche dei requisiti realizzate;
- liste di Identificatori Oggetti Omogenei, contenenti i nomi degli oggetti dello stesso tipo presenti nel sistema;
- liste di Informazioni derivate, contenenti i nomi degli attributi delle classi e i nomi dei tipi definiti e tipi ombrati presenti nel sistema.

E' permesso la creazione di un oggetto specificandone solo il nome. I requisiti di un oggetto vengono specificati, in qualsiasi ordine, aprendo le relative finestre sulla lista di Specifica Requisiti ad esso associata. Quando, nella specifica di un oggetto, viene fatto riferimento al nome di un oggetto non ancora definito, il sistema permette di creare un oggetto con quel nome.

I requisiti immessi vengono memorizzati in una base di dati chiamata Metadati. I controlli sulle definizioni dei requisiti immessi dall'analista vengono realizzati dai si-

stema in momenti diversi. Sono previsti tre tipi di controlli: di consistenza, di completezza e di adeguatezza.

Attualmente gli strumenti di supporto al Galileo/E sono in via di sviluppo.

3.0 RACCOLTA DEI REQUISITI.

Con la raccolta dei requisiti si vuole definire il comportamento aspettato del sistema da realizzare.

Definire il risultato di questa fase e' molto difficile e richiede molto tempo dell'intero processo di disegno.

Essa e' tuttavia la fase piu' importante perche' la maggior parte delle scelte successive nella fase di disegno sono basate su di essa. Di conseguenza la fase di raccolta e analisi dei requisiti ha un effetto a cascata su gli altri passi del processo di progettazione.

E' necessario descrivere tutte le informazioni utili per realizzare le successive fasi del processo di progettazione; tali informazioni sono ricavabili attraverso interviste, esame di documenti preesistenti ed osservazione diretta.

Il risultato di questa fase deve essere una descrizione coerente dei requisiti, che varra' utilizzata da categorie diverse di persone:

- gli utenti, che dovranno verificare la rispondenza con i loro bisogni;

- i progettisti, che dovranno realizzare il sistema;

- gli esperti, cioè coloro che provvederanno alla manutenzione e allo sviluppo del sistema realizzato.

3.1 ASPETTI TRATTATI NELLA RACCOLTA DEI REQUISITI.

È opinione diffusa che nel processo di progettazione di basi di dati occorra trattare aspetti strutturali, dinamici e quantitativi [Albano, 85a].

3.1.1 Aspetti strutturali.

Per quanto riguarda gli aspetti strutturali si identificano tre tipi di conoscenza:

- la conoscenza concreta, cioè i fatti specifici che si vogliono rappresentare: le entità, le loro caratteristiche e le associazioni tra entità;
- la conoscenza astratta, cioè i fatti generali che descrivono la conoscenza concreta rappresentata e limitano i modi in cui essa può evolvere;

- La conoscenza procedurale, cioè il modo in cui si opera sulla conoscenza concreta per modificarla o per ricavare altre informazioni con un procedimento di calcolo.

3.1.1.1 La conoscenza concreta

Con il termine "conoscenza concreta" si intendono i seguenti fatti:

1. le entità, cioè le cose che interessano di per sé;
2. le associazioni, cioè i fatti che stabiliscono un legame logico tra entità;
3. le proprietà, cioè quei fatti che interessano soltanto perché descrivono caratteristiche delle entità.

Per descrivere la conoscenza concreta si utilizzano i meccanismi di astrazione dei modelli semantici dei dati quali la classificazione, la aggregazione e la generalizzazione:

- classificazione.

Tramite questo meccanismo di astrazione entita' diverse vengono considerate omogenee in quanto appartenenti alla stessa categoria o classe.

Con la classificazione si astrae dalle differenze tra le singole entita' per evidenziare cio' che le accomuna .

Un importante vincolo di integrita' che presenta la classificazione e' che gli elementi di una classe sono tutti diversi tra loro.

- aggregazione.

E' il meccanismo di astrazione per definire la struttura degli elementi delle classi come aggregazione di componenti eterogenei.

La struttura e' descritta da un insieme finito di proprieta' comuni a tutti gli elementi.

Una proprieta' e' usata per definire un fatto significativo di entita' omogenee.

Con l'aggregazione si modellano sia i fatti descrittivi che hanno significato indipendentemente da altre entita' (attributi) sia le associazioni tra entita', cioe' i fatti che mettono in relazione entita' diverse.

- generalizzazione.

E' il meccanismo di astrazione per organizzare insieme di classi in una gerarchia definita da una relazione di ordinamento parziale: le classi in gerarchia modellano insieme di entita' ad un diverso livello di dettaglio.

Vengono distinte tre modalita' di definizione di sottoclassi: per sottoinsieme, per partizione, per restrizione.

Le sottoclassi "sottoinsieme", specializzazioni della stessa classe, non sono in generale tra di loro disgiunte, come lo sono invece le sottoclassi "partizione".

Le sottoclassi "restrizione" contengono gli elementi di una classe che hanno il valore di una o piu' proprieta' costanti in una relazione specificata.

In generale una sottoclasse puo' essere definita anche a partire da un'altra sottoclasse, modellando cosi' gerarchie a piu' livelli.

3.1.1.2 La conoscenza astratta.

Nel modellare una situazione reale, in generale non ci si limita a descrivere la conoscenza concreta, ma risulta necessario descrivere alcuni fatti generali che limitano i possibili valori della conoscenza concreta non riconducibili ai vincoli impliciti del modello dei dati [Rubenko, 30]

3.1.1.3 La conoscenza procedurale.

La conoscenza procedurale riguarda le informazioni dipendenti dal tempo e ricavabili a partire da ciò che è noto. Esse sono ottenibili tramite l'applicazione di funzioni (operazioni). Questo tipo di conoscenza può essere di aiuto per realizzare una modellizzazione di una situazione reale.

Le operazioni possono essere pensate come parte atomica delle procedure dell'organizzazione.

3.1.2. Aspetti dinamici.

Gli aspetti dinamici riguardano il modo in cui le procedure dell'organizzazione vengono svolte utilizzando la conoscenza procedurale e come esse interagiscono quando sono eseguite contemporaneamente.

Per descrivere il modo in cui evolve la base di dati per effetto dell'esecuzione delle procedure dell'organizzazione viene utilizzato il costrutto "attività" definito nel Galileo/8 [Albano,84]

Questa evoluzione della base di dati può avvenire per transazioni o per processi.

La semantica di una attività che modella una transazione può essere descritta sia mediante le "precondizioni" e "postcondizioni" sia in forma procedurale mediante il "corpo".

Si parla di evoluzione per processi quando le procedure dell'organizzazione sono di durata imprevedibile. Questo avviene quando le procedure sono in attesa del verificarsi di un evento o sono eseguite in parallelo con altre, con le quali possono interagire attraverso lo scambio di messaggi.

Per modellare questo aspetto dinamico è stato utilizzato il meccanismo dei "processi" del Galileo/8 [Albano,84] in

questo caso la semantica delle attivita' va descritta in modo procedurale.

La semantica di una attivita' che non modifica lo stato della base di dati puo' essere data soltanto in modo procedurale perche' le pre e post condizioni risultano essere, in questo caso, prive di senso.

3.1.3 Aspetti quantitativi.

Per le fasi successive della progettazione e' opportuna la specifica di alcuni parametri quantitativi, come ad esempio la cardinalita' di una classe o di una associazione, la volatilita' e la probabilita' di una associazione, la frequenza di esecuzione di una operazione e di una attivita'.

3.2 DEFINIZIONE DEI REQUISITI.

Nel caso in studio le classi della realta' che si vuole modellare vengono pensate come elementi di altre classi,

le metaclassi, per astrarre proprietà comuni a più classi.

Sono state individuate le seguenti metaclassi: Progetti, settori, Attività, Operazioni, Classi, Attributi, Associazioni, Tipi [Navazio, 85]

In appendice A viene riportato lo schema concettuale della database di dati realizzata.

3.2.1. La definizione dei progetti e dei settori.

La specifica dei requisiti di un sistema informatico consiste, in generale, nella specifica dei requisiti di ciascuno dei settori dell'organizzazione aziendale in cui il sistema verrà inserito; i requisiti di ogni sistema fanno parte di un progetto. I requisiti di un progetto sono:

1. il nome del progetto;
2. una descrizione in linguaggio naturale del progetto;
3. il responsabile, cioè la persona responsabile dell'evoluzione del progetto;

4. la password, cioè un codice di riconoscimento del responsabile del progetto;

5. la data d'inizio e quella di ultima modifica, cioè la data in cui ha inizio la stesura del progetto e la data dell'ultima modifica del progetto.

I requisiti di un settore sono:

1. il nome del settore;

2. una descrizione in linguaggio naturale del settore;

3. la data dell'ultima modifica dei requisiti raccolti nel settore.

3.2.2 La definizione dei dati.

I dati di ogni settore vengono modellati dalle classi. E' possibile definire anche sottoclassi di tipo sottoinsieme, partizione e restrizione. I requisiti sulle classi sono:

1. il nome della classe;
2. una descrizione in linguaggio naturale che specifica il significato della classe;
3. gli utenti dell'organizzazione che forniscono i requisiti sulla classe;
4. la cardinalita' della classe, cioe' il numero di elementi della classe. E' possibile specificare il valore minimo, massimo, medio ed il tasso di crescita annuo;
5. i vincoli d'integrita' che vengono modellati con il meccanismo delle asserzioni e specificati in linguaggio naturale. Un particolare vincolo predefinito e' quello delle chiavi. Gli attributi chiave non possono essere di tipo opzionale;
6. gli attributi della classe. Per ogni attributo si specifica il nome, una descrizione della proprieta' modellata dall'attributo, il tipo dei valori dell'attributo, alcuni parametri quantitativi. Eventuali informazioni sulla distribuzione dei valori dell'attributo possono essere espressi informalmente nella descrizione dell'attributo.

Per gli attributi di tipo opzionale si puo' specificare la probabilita' di esistenza;

7. le associazioni di cui la classe e' dominio. Per ogni associazione si specificano il nome, le caratteristiche, una descrizione informale, il nome dell'associazione inversa, la classe in cui e' definita l'inversa e i parametri quantitativi.

Le caratteristiche della associazione sono la totalita', la molteplicita', la modificabilita'.

I parametri quantitativi che si possono specificare su una associazione sono: la cardinalita', cioè il numero di elementi della classe codominio che sono in associazione con uno stesso elemento della classe dominio (questo parametro ha senso soltanto per le associazioni multiple), e' possibile specificare il valore minimo, massimo, medio e il tasso di crescita annuo; la probabilita' di esistenza che da' una indicazione sul numero di elementi della classe dominio che hanno associato almeno un elemento della classe codominio (questo parametro ha senso soltanto per le associazioni parziali); la volatilita', cioè la frequenza con cui vengono cambiati gli elementi associati.

3.2.3 La definizione dei tipi.

L'utente puo' definire dei tipi da utilizzare nella specifica del tipo degli attributi delle classi.

I requisiti che si possono specificare sui tipi definiti dall'utente sono:

1. il nome del tipo;
2. una descrizione informale della semantica del tipo;
3. vincoli d'integrita' che possono essere espressi con il meccanismo delle asserzioni. Le asserzioni vengono date in linguaggio naturale;
4. la data dell'ultima modifica dei requisiti del tipo definito.

3.2.4 La definizione delle operazioni.

L'utente puo' definire delle operazioni sulle quali possono essere specificati i seguenti requisiti:

1. il nome della operazione;
2. una descrizione informale della semantica della operazione;
3. la frequenza con cui viene eseguita l'operazione;
4. la data dell'ultima modifica dei requisiti dell'operazione;
5. le precondizioni, cioè delle condizioni date sulle classi che devono valere prima che inizi l'esecuzione della operazione;
6. le postcondizioni, cioè delle condizioni date sulle classi che devono valere quando termina l'esecuzione della operazione;
7. le eccezioni, cioè particolari situazioni di errore che si possono verificare durante l'esecuzione della operazione;
8. il corpo della operazione. La specifica di esso può essere una alternativa alla specifica fatta con le pre(post) condizioni. Comunque, se l'utente lo ritiene significativo, la semantica di una operazione può essere fatta con entrambe le modalità.

3.2.5 La definizione delle attività.

Le attività vengono utilizzate per modellare le procedure svolte in un certo settore aziendale. L'utente non vede le operazioni ma soltanto le attività.

I requisiti che si specificano su una attività sono:

1. il nome della attività;
2. una descrizione in linguaggio naturale che specifica la procedura aziendale modellata dalla attività;
3. gli utenti dell'organizzazione che hanno fornito i requisiti sull'attività;
4. il livello dell'organizzazione a cui viene svolta la procedura: può essere strategico, gestionale o operativo;
5. il settore aziendale in cui viene espletata la procedura modellata dall'attività;
6. la frequenza di esecuzione dell'attività;

7. la modalita' di esecuzione dell'attivita': puo' essere "batch" oppure "online";
8. la priorita' di esecuzione dell'attivita' rispetto alle altre: puo' essere alta, media oppure bassa;
9. il tempo di attesa tollerato dall'utente per eseguire l'attivita'. Si puo' specificare il tempo massimo e quello medio.
La specifica di questo parametro ha senso solo se l'attivita' modella una transazione e non nel caso in cui modella un processo;
10. la data dell'ultima modifica dei requisiti dell'attivita';
11. le precondizioni, cioè delle condizioni date sulle classi che devono valere prima che inizi l'esecuzione dell'attivita';
12. le postcondizioni, cioè delle condizioni date sulle classi che devono valere quando termina l'esecuzione dell'attivita';
13. le eccezioni, cioè particolari situazioni di errore che si possono verificare durante l'esecuzione dell'attivita';

14. il corpo dell'attività'. La specifica di esso può essere una alternativa alla specifica fatta con le pre(post) condizioni. Comunque, se l'utente lo ritiene significativo, nella definizione di una attività è consuetudine l'uso di entrambe le modalità. In particolare la specifica procedurale appare necessaria quando l'attività modella un processo o una transazione che non modifica lo stato della base di dati.

3.3 CONTROLLI.

La metabase di dati evolve secondo una sequenza di transazioni tra i seguenti possibili stati:

1. stato consistente: le definizioni dei requisiti memorizzate nella metabase soddisfano i vincoli imposti dal modello utilizzato per la specifica;
2. stato completo: la specifica dei requisiti contiene tutte le informazioni necessarie agli oggetti definiti nella metabase;

3. stato adeguato: tutte le informazioni memorizzate nella metabase sono rilevanti per la definizione di un modello del comportamento aspettato del sistema da realizzare.

La corretta evoluzione della metabase e' garantita da controlli che il sistema esegue sulle definizioni immesse dall'utente. Sono state individuate tre classi di controlli: di consistenza, di completezza e di adeguatezza. Le caratteristiche del sistema e il modo di operare di ogni tipo di controllo ne consigliano l'attivazione in momenti diversi della specifica.

1.3.1 Controlli di consistenza.

Questi controlli assicurano che le definizioni immesse dall'utente soddisfino i vincoli imposti dal modello utilizzato.

Si distinguono tre tipi di controlli di consistenza a seconda del momento in cui essi vengono eseguiti:

1. controlli eseguiti durante la fase di edizione delle definizioni;

2. controlli eseguiti al momento della memorizzazione di una definizione;
3. controlli eseguiti su esplicita richiesta dell'utente.

3.3.2 Controlli di completezza.

Il sistema realizzato offre all'utente la possibilita' di definire i requisiti in maniera incrementale.

L'unico vincolo imposto dal sistema nel creare un nuovo oggetto nella metabase e' quello di definire il suo nome (il settore in cui e' attivo viene impostato dal sistema) permettendo quindi di specificare in maniera piu' dettagliata le caratteristiche dell'oggetto al momento ritenuto opportuno dall'utente.

Cio' comporta il fatto che la metabase puo' contenere definizioni incomplete.

Per poter eseguire successivamente controlli di adeguatezza, e' necessario che la specifica di certi requisiti sia presente.

I controlli di completezza vengono realizzati su specifica richiesta dell'utente.

1.3.3 Controlli di adeguatezza.

L'immissione, la modifica e la cancellazione interattiva può portare ad avere nella metabase degli oggetti che non sono necessari per definire un modello della realtà.

Ciò può succedere che uno stato finale della metabase, inteso come possibile rappresentazione completa delle specifiche dell'utente, abbia degli oggetti che non hanno alcuna relazione logica con la specifica. Ad esempio, può esistere una classe che non è utilizzata da nessuna attività; occorre che ogni tipo definito dall'utente sia utilizzato per tipizzare almeno uno dei seguenti oggetti : un parametro di una attività e di una operazione, il risultato di una operazione oppure un attributo di una classe.

I controlli di adeguatezza vengono eseguiti su esplicita richiesta dell'utente.

Gli errori di adeguatezza sono di natura logica e dipendono dalla realtà che si vuole modellare e non dai contenuti della metabase.

4.0 ARCHITETTURA DEL SISTEMA.

4.1 FUNZIONALITA' DI UNO STRUMENTO AUTOMATICO.

Una problematica della progettazione di basi di dati e' dovuta al fatto che passa troppo tempo dal momento in cui si raccolgono i requisiti al momento in cui si dispone di una prima versione funzionante della realizzazione, con conseguente difficolta' ad apportare delle modifiche ai requisiti.

La specifica dei requisiti per una grande organizzazione puo' comportare notevoli difficolta' con eccessiva impossibilita' di effettuare manualmente controlli di consistenza e compatibilita'.

A causa della riconosciuta difficolta' nell'espressione dei propri bisogni incoerenti da parte dell'utente, coloro che realizzano il sistema informativo avvertono la necessita' di una metodologia per definire questi bisogni in maniera sistematica. Questo processo inizia con l'analisi della realta' in studio e termina con il fornire una specifica formale dei requisiti dell'utente.

I principali vantaggi che l'approccio automatico presenta rispetto alla gestione manuale dei requisiti sono:

1. la qualita' della specifica:
gestire manualmente la specifica puo' comportare ambiguita', mentre una gestione automatica e' piu' precisa in quanto la descrizione viene forzata secondo prefissate "linee guide";
2. il controllo di ridondanza:
con una gestione manuale diversi documenti contengono le stesse informazioni, talvolta con significati diversi; cio' puo' portare a delle inconsistenze, mentre l'utilizzo del calcolatore tende ad eliminare la presenza di ridondanze e ambiguita';
3. la completezza della descrizione:
con una gestione manuale si avvertono difficolta' a realizzare controlli di correttezza e completezza che invece risultano piu' agevoli con l'utilizzo del calcolatore;
4. la semplicita' di manutenzione:
una gestione manuale rende complesse le modifiche nel corso del progetto e cio' puo' provocare una rapida degradazione della rispondenza del progetto con il sistema reale che descrive, mentre uno strumento automatico consente modifiche rapide e affidabili.

4.2. SCELTA DELL'AMBIENTE HARDWARE/SOFTWARE.

Nella identificazione dell'ambiente hardware/software hanno influito in particolare le seguenti considerazioni:

1. la portabilità: possibilità di poter operare su diversi tipi di computer e quindi di girare su diversi sistemi operativi;
2. la possibilità di avere un prodotto da poter utilizzare sia su mainframe che su personal computer;
3. l'inclusione di altri programmi scritti in altri linguaggi;
4. la possibilità di poter utilizzare le funzionalità che offre l'editore di sistema;
5. la possibilità di avere una facile interazione con il prodotto. Questo aspetto è dovuto al tipo di utenza prevista;
6. la possibilità di poter ottenere rapporti di vario tipo;

7. l'opportunità di poter definire a programma delle forme video;
8. la facilità nel gestire e sviluppare le applicazioni.

4.3 IL FOCUS: LINGUAGGIO DELLA IV GENERAZIONE.

In [Martia, 82] vengono esaminate alcune caratteristiche di alcuni nuovi strumenti software individuando un denominatore comune: la possibilità di svolgere in tempi estremamente brevi le attività svolte con gli strumenti di programmazione tradizionali.

Gli strumenti software che hanno questa peculiarità vengono detti linguaggi della IV generazione.

Non è l'arco di utilizzo né l'incremento di produttività che essi consentono che li definisce. Infatti, possono essere considerati linguaggi della IV generazione:

- linguaggi di tipo general purpose e specialistici;
- linguaggi di tipo procedurale e non procedurale;

- linguaggi di query o report generator;
- linguaggi per realizzare grafici o statistiche (ad esempio il SAS).

E' opportuno notare come da questo tipo di definizione segua che questi strumenti molto spesso non siano sovrapponibili, e quindi neanche direttamente confrontabili, per via delle caratteristiche peculiari che ognuno di essi presenta.

Il FOCSS e' un sistema integrato per la creazione di basi di dati e per il controllo delle informazioni in essi memorizzate, e viene presentato come uno strumento software ad alta produttivita' per lo sviluppo di applicazioni grazie ai suoi moduli di gestione ed interrogazione di basi di dati, prototipizzazione delle applicazioni, di statistica, grafica e comunicazione.

4.3.1. L'Ambiente.

Il FOCSS e' stato disegnato per funzionare interattivamente su IBM o elaboratori (Mainframe) compatibili sotto sistemi operativi quali VS/CMS o MVS con TSO, CICS

e/o IMS/DC. Esso può anche funzionare in modo batch su ogni sistema operativo OS.

Esiste anche una versione del FOCUS (PC/FOCUS) disegnata per funzionare interattivamente su personal computer IBM e compatibili in ambiente MS/DCS. Le procedure FOCUS scritte in un certo ambiente possono essere trasportate in altri ambienti.

In tutti i casi, e' possibile, senza terminare la sessione FOCUS, accedere a tutti i servizi offerti normalmente dal sistema operativo, premettendo al comando la sigla che identifica il particolare sistema operativo (CMS, ISO, DCS).

E' possibile interfacciare diversi linguaggi di programmazione, oppure richiedere i servizi del FOCUS da un linguaggio ospite (ASSEMBLER, PL/I, COBOL, FORTRAN).

Le procedure di utente possono essere lette in linea o memorizzate in opportune librerie e richiamate. In ogni caso, le procedure utente vengono interpretate dal FOCUS e non richiedono passi di compilazione separati prima dell'esecuzione.

4.3.2 Definizione della base di dati.

Il modello dei dati cui fanno riferimento le strutture FOCUS e' quello gerarchico.

E' possibile definire delle strutture gerarchiche in un sottorambiente FOCUS, caratterizzato dal fatto che i file in esso definiti presentano sempre lo stesso tipo (MASTER).

Il FOCUS non presenta una architettura a tre livelli di descrizione dei dati; questo comporta il coglobare nella stessa definizione sia aspetti logici che aspetti fisici.

La descrizione di una base di dati gerarchica, memorizzata in un unico archivio fisico, ha la seguente struttura:

```
FILENAME=filename, SEFFIN=FOC(,S)
{SEGNAME=segmentname(, PARENT=segmentname)}
{FIELDNAME=fieldname(, ALIAS=alternativefieldname)+}
```

Le parole scritte in maiuscolo (parole chiave) sono dei comandi che accettano un certo numero di parametri. Nella struttura data sopra si e' tenuto conto solo degli aspetti logici, riservandoci di trattare quelli fisici in un secondo momento.

- FILENAME=filename e' utilizzato per dare un nome simbolico al file che si vuole creare;
- SEGNAME=segmentname e' utilizzato per definire un nodo dell'albero. Viene specificato il nome del nodo o il nome del padre (usando il parametro PARENT, non specificato quando si definisce la radice);
- FIELDNAME=fieldname e' utilizzato per definire un campo di un segmento. E' possibile definire un nome alternativo specificando un valore per il parametro ALIAS. In tutti i casi e' possibile individuare il campo con un qualunque troncamento univoco. Si definisce il formato del campo specificando il parametro FORMAT, permettendo che il campo possa assumere valori numerici o alfanumerici.

L'unico aspetto fisico che e' possibile specificare in FOCUS e' quello relativo alla costruzione di un indice su un campo specificando il parametro FIELDTYPE=I.

Aspetti riguardanti l'organizzazione fisica dell'archivio che contiene la base di dati che si vuole definire sono trasparenti all'utente.

La descrizione di un file può definire sia la struttura logica di una base di dati FOCUS che di una base di dati esterna. In quest'ultimo caso il parametro SUFFIX può assumere uno dei seguenti valori : IIX, CCM, ISAM, VSAM,

program name (e' il nome del modulo eseguibile dell'interfaccia).

L'accesso ad un file esterno e' consentito solo in lettura. Sono disponibili interfacce per la lettura di files sequenziali, ISAM, VSAM, e per i DBMS piu' diffusi (IMS, ADADAS, ICITAI, SYSTEM 2000, e altri).

In tutti questi casi, occorre descrivere al FOCUS la base di dati esterna come una struttura gerarchica. Poiche' in generale la sintassi di definizione degli archivi FOCUS non e' abbastanza ricca da descrivere in modo completo la struttura dei files esterni, un certo numero di informazioni viene memorizzato su un altro file, denominato ACCESS file.

Di conseguenza, la struttura della base di dati esterna, e le strategie di navigazione, sono determinate in base alle informazioni contenute nel MASTER e nell' ACCESS file.

E' disponibile una procedura FOCUS che verifica la correttezza sintattica dello schema logico-fisico (BESTFILE filename).

4.2.3 Manipolazione dei dati.

Il sottosistema dedicato a queste scoppe fornisce un linguaggio che permette di effettuare sulla base di dati operazioni di: modifica di records esistenti, aggiunta di nuovi records, cancellazione di records esistenti. Questo linguaggio e' di tipo non procedurale e con una limitata struttura del controllo, perche' previsto per essere facilmente utilizzabile interattivamente, anche da persone non esperte.

L'interazione con la base di dati avviene per transazioni. I dati della transazione possono essere immessi sia direttamente da terminale (e' permessa anche una interazione di tipo full-screen) che da un file su cui siano stati precedentemente memorizzati. Una transazione presenta la seguente struttura:

```
MODIFY FILE filename  
componente1  
{componentei}  
END
```

Ogni componente permette di:

1. Descrivere il formato dei dati della transazione

I sottocomandi disponibili sono:

a. FIRMFORM, con cui si specifica che i dati di input sono in formato fisso

b. FBSEFORM, con cui si specifica che i dati di input sono in formato libero, cioè espressi come:

fieldname = fieldvalue

c. FRCHEP specifica che i valori richiesti dalla transazione vengano immessi dietro richiesta del sistema

3. CHEFORM permette di definire uno schermo logico, che si può espandere anche su più schermi fisici. In tal caso il FOCUS interpreta i tasti funzionali F26 e F27 come comandi di paginazione (rispettivamente cose pagina avanti e pagina indietro).

2. **Descrivere i criteri di validazione per accettare i dati della transazione**

Il sottocomando da utilizzare è VALIDATE.

Il test di validazione è un calcolo eseguito su un'espressione e può assumere solo due valori: vero o falso. Se il risultato è falso (il valore assunto è 0) la transazione è rifiutata come invalida,

altimenti e' accettata. Dopo il test e' possibile usare il sottocomando CN INVALID seguito da TYPE o GOTD.

3. Effettuare elaborazioni sui valori della transazione

Il sottocomando da utilizzare e' CCMHPE.
Serve per assegnare valori a dei campi di memoria predefiniti nel MASTER o all' interno della procedura. Se il campo e' nella descrizione del MASTER viene assunto il formato gia' dichiarato nel MASTER stesso. Se il campo non e' dichiarato nel MASTER, deve essere definito anche il formato.

4. Memorizzare l'insieme dei valori su cui si vuole operare

I sottocomandi da utilizzare sono:

a. MATCH e' l'operazione di base durante una transazione.

Consiste nel confrontare alcuni valori della transazione con i corrispondenti valori della base di dati

b. NEXT serve per localizzare una delle successive istanze logiche del segmento.

5. Descrivere le azioni da intraprendere una volta individuato l'insieme su cui si vuole operare

In seguito all'esecuzione di un comando MATCH e' possibile specificare le azioni da svolgere utilizzando il sottocomando CN NCMATCH o il sottocomando CN MATCH. Per quest'ultimo e' possibile specificare i seguenti parametri:

- CONTINUE: nessuna azione viene compiuta su quel segmento, ma si procede al successivo comando MATCH
- INCLUDE: include le istanze del segmento e dopo tale azione inizia il processo della successiva transazione
- REJECT: rifiuta quella transazione e non viene eseguita nessuna azione su quel segmento, ma si passa alla successiva transazione
- DELETE: cancella le istanze di quel segmento e tutte le istanze discendenti, dopodiche' comincia ad eseguire la successiva transazione. E' sufficiente eseguire il MATCH solo sul campo

chiave per cancellare tutte le istanze di quel segmento

- UPDATE: serve per cambiare le istanze dei campi non chiave di quel segmento.

E' possibile effettuare delle elaborazioni utilizzando il sottocomando COMPUTE (vedi il punto 3), e controllare la logica dell'applicazione con i comandi GOTO e IF (vedi il punto 7). ON NOMATCH prevede le stesse azioni di ON MATCH, escluso CONTINUE, DELETE e UPDATE.

In seguito all'esecuzione di un comando NEXT e' possibile specificare le azioni da svolgere utilizzando il sottocomando ON ONNEXT o il sottocomando ON NEXT. ON NEXT prevede le stesse azioni di ON MATCH e ON ONNEXT prevede le stesse azioni di ON NOMATCH.

6. Descrivere la sorgente delle transazioni

- a. START N dove N e' il numero della transazione dalla quale il processo dovra' cominciare
- b. STOP N dove N e' il numero della transazione dopo la quale il processo dovra' finire
- c. DATA e' l'ultimo sottocomando che deve essere sempre inserito per indicare la fine della lista dei sottocomandi. Si puo' utilizzare sotto le seguenti forme:

- DATA quando viene utilizzato il EBCDIC
- DATA ON *ddname* quando i dati sono reindirizzati su un file esterno (CMS/FILEDEF TSO/ALLOD)
- DATA VIA E13270/FIDEL quando la sorgente dei dati e' una forma video (schermo formattato).

7. Controllare la logica dell' applicazione.

Sono disponibili a questo scopo alcuni comandi:

- a. CASE permette di combinare molte routines "modify" all'interno di un' unica routine e di saltare dall' una all' altra dopo un opportuno comando di GOTO o di IF. Ciascun case e' seguito da una label che non deve superare i dodici caratteri. TOP e ENY sono nomi riservati
- b. ENDCASE segnala la fine di ciascun "case" logico
- c. IF (espressione) GOTO: viene valutata l' espressione e se risulta vera viene eseguito il salto altrimenti viene valutato il seguente IF oppure si passa al successivo sottocomando
- d. GOTO salto incondizionato; se all' interno di un CASE non vi e' un trasferimento esplicito ad un altro CASE viene assunto GOTO TOP. In particolare GOTO ENY permette di terminare la procedura.

E' comunque permesso di operare su basi di dati PCCBS attraverso programmi scritti in ogni linguaggio per cui

e' prevista l'interfaccia (CCSCL, SL/1, FORTRAN e altri). I servizi vengono richiesti dal linguaggio ospite mediante il proprio meccanismo standard di attivazione di sottoprogrammi utilizzando una procedura predefinita di nome POCUS.

Viene di seguito mostrato un esempio di definizione di una base di dati FOCUS:

```

FILENAME=PROVA, SUFFIX=FOC
SEGNAME=A, SEGTYPE=S1
  FIELD=A1, ALIAS=AUNC, FORMAT=A30, F
  FIELD=A2, ALIAS=ADUE, FORMAT=A30, F
SEGNAME=B, PARENT=A, SEGTYPE=S1
  FIELD=B1, ALIAS=BUNC, FORMAT=A30, F
  FIELD=B2, ALIAS=BDUE, FORMAT=A30, F
SEGNAME=C, PARENT=A, SEGTYPE=S1
  FIELD=C1, ALIAS=CBNO, FORMAT=I6, F
  FIELD=C2, ALIAS=CDUE, FORMAT=A30, F
SEGNAME=D, PARENT=B, SEGTYPE=S1
  FIELD=D1, ALIAS=DNOC, FORMAT=I6, F
  FIELD=D2, ALIAS=DDUE, FORMAT=A10, F

```

Questa base di dati definisce la seguente struttura:

```

      A
    G1  S1
*****
*A1      **
*A2      **
*        **
*****
      I
    +-----+
    I          I
    I B          I C
02  I S1      04  I S1
*****
*B1      **   *C1      **
*B2      **   *C2      **
*        **   *        **
*****
      I
    I J
    03  I S1
*****
*D1      **
*D2      **
*        **
*****

```

I successivi esempi descrivono l'utilizzo del linguaggio di manipolazione dei dati del POCUS in relazione ad operazioni di inclusione, modifica e cancellazione relative alla base di dati sopra descritte:

1. Es1: Si vuole inserire l'istanza dieci(E1) del segmento B come figlia dell'istanza quattro(A1) del segmento A supponendo che il campo A1 contenga tale valore.

```

MODIFY FILE PROVA           { Definisce l'ambiente di modifica
                             per il file PROVA }
CREATE
" Campo A1 : <A1 "        { Acquisizione dei dati }
" Campo E1 : <E1 "
MATCH A1                   { Esegue una lettura diretta sulla B.D.
                             PROVA per il valore "quattro" di A1 }
ON NOMATCH REJECT         { Se l'istanza di B non esiste rifiuta
                             la transazione }
ON MATCH CONTINUE         { Se l'istanza di A esiste continua
                             per verificare l'esistenza della
                             istanza di B }
MATCH B1                   { Esegue una lettura diretta sulla B.D.
                             PROVA per il valore "dieci" di E1 come
                             figlio del valore "quattro" di A1 }
ON NOMATCH INCLUDE        { Se l'istanza di B non esiste la
                             include nella E.D. }
ON MATCH REJECT           { Se l'istanza di B esiste gia' rifiuta
                             la transazione }
DATA VIA TIB270           { Definisce il supporto di INPUT per la
                             acquisizione dei dati }
END                         { Termina definizione dell'ambiente
                             relativo alla E.D. PROVA }

```

2. Es2: Si vuole modificare l'istanza dieci(E1) del segmento B, per il valore B2, come figlia dell'istanza quattro(A1) del segmento A supponendo che i campi E1 e A1 contengano i valori dati in precedenza. Nel caso che E1 non esista la include.

```

MODIFY FILE PROVA          | Definisce l'ambiente di modifica
                           | per il file PROVA
CRIFCR4                   | Acquisizione dei dati
" Campo A1 : <A1 "
" Campo B1 : <B1 "
" Campo E2 : <E2 "

MATCH A1                   | Esegue una lettura diretta sulla B.D.
                           | PROVA per il valore "quattro" di A1
ON NOMATCH REJECT        | Se l'istanza di A non esiste rifiuta
                           | la transazione
ON MATCH CONTINUE        | Se l'istanza di A esiste continua
                           | per verificare l'esistenza della
                           | istanza di E
MATCH B1                   | Esegue una lettura diretta sulla B.D.
                           | PROVA per il valore "dieci" di B1 come
                           | figlio del valore "quattro" di A1
ON NOMATCH INCLUDE       | Se l'istanza di E non esiste
                           | la include nella B.D.
ON MATCH UPDATE E2       | Se l'istanza di E esiste gia'
                           | modifica il corrente valore di E2
DATA VIA FIB270          | Definisce il supporto di INPUT per la
                           | acquisizione dei dati
END                       | Termina definizione dell'ambiente
                           | relativo alla B.D. PROVA

```

3. Es3: Si vuole cancellare l'istanza dieci (E1) del segmento 2 come figlia dell'istanza quattro (A1) del segmento A supponendo che i campi E1 e E1 contengano i valori dati in precedenza.

```

MODIFY FILE PROVA      ( Definisce l'ambiente di modifica
CRIFORE               ( Acquisizione dei dati )
" Campo A1 : <A1 "
" Campo B1 : <B1 "

MATCH A1              ( Esegue una lettura diretta sulla E.D.
                       EBOVA per il valore "quattro" di A1 )
OR NOMATCH REJECT    ( Se l'istanza di A non esiste rifiuta
                       la transazione )
ON MATCH CONTINUE    ( Se l'istanza di A esiste continua
                       per verificare l'esistenza della
                       istanza di B )
                       )
JAJON B1              ( Esegue una lettura diretta sulla E.D.
                       PROVA per il valore "dieci" di E1 come
                       figlio del valore "quattro" di A1 )
ON NOMATCH REJECT    ( Se l'istanza di B non esiste rifiuta
                       la transazione )
ON MATCH DELETE      ( Se l'istanza di B esiste già
                       la cancella. )
DATA VIA F11270      ( Definisce il supporto di INPUT per la
                       acquisizione dei dati )
END                  ( Termina definizione dell'ambiente
                       relativo alla E.D. PROVA )

```

4.3.4. Funzionalità del FOCDS.

Per funzionalità in questo contesto si intendono quelle funzionalità che qualificano un Sistema di Gestione di Basi di Dati e che dovrebbero essere presenti in ogni

prodotto di questa categoria [Albano, #5a] : indipendenza logica e fisica, controllo dei dati.

- **indipendenza logica e fisica**

-- Per indipendenza (delle applicazioni dall'organizzazione) logica (dei dati) si intende la possibilità che i programmi applicativi non debbano essere modificati in seguito a modifiche dello schema logico, per aggiunta di nuove definizioni e per modifiche o eliminazione di quelle esistenti.

E' possibile riorganizzare uno schema logico-fisico in ambiente RDCIS utilizzando il comando REBUILD; specificandone opportuni parametri e' possibile:

1. aggiungere nuovi campi in fondo a segmenti gia' esistenti;
2. rimuovere campi;
3. cambiare l'ordine dei campi in un segmento;
4. aggiungere nuovi segmenti come discendenti di altri gia' esistenti;

5. ricavare segmenti;

6. costruire degli indici su campi.

Le applicazioni che non hanno alcun riferimento alle variazioni fatte sullo schema logico non subiscono alcuna modifica.

-- Per indipendenza (delle applicazioni dall'organizzazione) fisica (dei dati) si intende la possibilità che i programmi applicativi non debbano essere modificati in seguito a modifiche dell'organizzazione fisica dei dati.

Per costruzione il SOCCS realizza in modo abbastanza soddisfacente questo tipo di indipendenza. Gli operatori disponibili agli utenti non dipendono dall'organizzazione fisica dei dati.

Gli aspetti riguardanti l'organizzazione fisica dell'archivio che contiene la base di dati sono trasparenti all'utente.

- controllo dei dati

Si intendono i seguenti aspetti:

-- integrità

Esiste un modulo FOCUS che controlla che i dati inseriti e modificati siano conformi alle definizioni. Quando si cancellano tutte le istanze di un segmento in una base di dati, una parte del sistema FOCUS provvede a cancellare tutte le istanze dei segmenti del sottoalbero che ha come radice il segmento in questione.

-- affidabilità

Utilizzando il comando CHECK in ambiente MODIFY è possibile specificare dopo quante transazioni si vuole modificare lo stato della base di dati. È possibile realizzare una copia della base di dati da utilizzare nel caso in cui si verificano malfunzionamenti sia hardware che software.

Utilizzando il comando LOG è possibile visualizzare tutte le transazioni rifiutate, e mercantizzare su un file tutte le transazioni accettate. È possibile anche distinguere le transazioni rifiutate in funzione dei motivi del rifiuto.

-- sicurezza (autorizzazione e protezione)

Per ogni base di dati FOCUS può essere individuato un utente privilegiato con funzioni di DBA. Il DBA può controllare l'accesso ai dati, sia per restringere i dati accessibili sia per

limitare le operazioni che si possono fare su di essi. E' possibile accedere in sola lettura, in sola scrittura, in lettura e scrittura, specificando nella definizione della base di dati per il comando ACCESS rispettivamente i valori R, W, e RW.

Per controllare che ai dati accedano solo persone autorizzate il POCBS permette di specificare delle parole di riconoscimento (PASSWORD).

Inoltre e' possibile restringere i valori accessibili specificando opportuni parametri per il comando SECURITY.

E' possibile anche crittografare la base di dati e lo schema logico-fisico.

5.0 IL SISTEMA REALIZZATO.

5.1 L'INTERFACCIA UOMO-SISTEMA.

Nella realizzazione dell'interfaccia utente-sistema ci si e' posto l'obiettivo di ottenere un'agevole interazione per l'utente. Si e' ritenuto opportuno disegnare un'interfaccia in cui l'interazione fosse basata sulla compilazione o modifica di "moduli" o "form" predisposti allo scopo di raggruppare i dati secondo i criteri che hanno portato alla definizione della struttura delle metaclassi.

I dati di una qualsiasi applicazione vengono cosi' immessi attraverso un'interazione di tipo "full_screen".

Viene di seguito data una breve descrizione delle PRKEY:

- PF01 FOCUS: ritorno all'ambiente FOCUS.
- PF02 UNDO: ripristina la mappa del video al momento in cui e' stata chiamata. E' utile se, dopo aver effettuato alcune modifiche, si vogliono ripristinare i vecchi valori.
- PF03 FOCUS: ritorno all'ambiente FOCUS.
- PF04 END: ritorno al form logicamente precedente quello corrente.
- PF06 DELETE: cancella dalla base di dati i dati contenuti nel form logico presente sul video.
- PF07 BACKWARD: visualizza la mappa precedente quando un form occupa piu' di una mappa dello schermo.
- PF08 FORWARD: visualizza, a differenza della precedente, la mappa successiva quando un form occupa piu' di una mappa dello schermo.
- " HELP: attiva l'help che da' una descrizione delle caratteristiche di ciascun campo relativi-

vo al foro su cui si sta operando: tipo, lunghezza, descrizione.

~~ENTER~~ CONTINUE: effettua una transazione.

Prenderemo ora in considerazione il modo in cui devono essere definiti i requisiti. Si e' ritenuto opportuno che l'utente non debba essere obbligato a procedere secondo una metodologia prestabilita. Da questo punto di vista, l'unico vincolo imposto e' quello di creare un settore ed un progetto corrente prima di poter definire altri tipi di oggetti (classi e proprieta', tipi, operazioni, attivita').

E' da sottolineare il fatto che per definire un qualsiasi oggetto e' sufficiente specificarne il nome.

Per quanto riguarda il sistema di supporto realizzato, non solo sono considerate valide le definizioni incomplete, ma sono anche ammesse le definizioni implicite: se l'utente fa riferimento ad oggetti non ancora definiti, il sistema li crea automaticamente dietro esplicita conferma dell'utente.

Dunque l'utente puo' dare le definizioni nel modo che ritiene piu' opportuno, puo' effettuare una graduale immissione delle definizioni ed in qualsiasi punto dell'applicazione puo' fare ritorno al foro logicamente

precedente quello corrente (PF04) e terminare definitivamente facendo ritorno all'ambiente ICCUS (PF01/PF03).
E' disponibile un "HELP online" (PF08) che descrive le caratteristiche degli attributi delle metaclassi.

5.1.1 Navigazione nel sistema.

Il primo form accessibile del sistema realizzato permette la specifica del nome di un progetto. I riferimenti alle figure di seguito specificati si riferiscono all'escapio di sessione descritte nel paragrafo 5.3.

I progetti:

Una volta specificato il nome di un progetto (vedi figura 2) il sistema ne verifica l'esistenza nella metabase.

Se il progetto non e' presente ne viene permessa l'inserzione tramite il relativo form che visualizza gli attributi della metaclassa "progetto" (vedi figura 3). Se il progetto specificato esiste si entra nel form che da' la possibilita' di modificarne il nome; dopodiche' il pannello successivo visualizza, in una forma analoga a quella utilizzata per l'inserzione, il contenuto della metaclassa "progetto" relativamente al nome specificato.

A questo punto si ha la possibilità di apportare delle modifiche ai valori precedentemente memorizzati oppure di cancellare il progetto corrente. La modifica del nome di un progetto si riflette nelle eventuali informazioni ad esso connesse. La cancellazione di un progetto implica la cancellazione di tutte le informazioni ad esso logicamente connesse.

Se l'utente opera su un progetto già esistente cancellandolo o aggiornandone il nome, il sistema ripropone il form iniziale relativo alla specifica del nome di un progetto.

Se l'utente inserisce un nuovo progetto o aggiorna i dati, ma non il nome, relativi ad un progetto già esistente, il sistema visualizza il form che permette di specificare il nome di un settore relativamente al progetto corrente (vedi figura 7).

I settori:

Una volta specificato il nome di un settore il sistema ne verifica l'esistenza nella metabase relativamente al progetto corrente.

Se il settore non c'è presente ne viene permessa l'inserzione tramite il relativo form che visualizza gli attributi della metaclassa "settore" (vedi figura 8). Se il settore specificato esiste si entra nel form che dà la possibilità di modificarne il nome; dopodiché il pannello successivo visualizza, in una forma analoga a

quella utilizzata per l'inserzione, il contenuto della metaclassa "settore" relativamente al nome specificato. A questo punto si ha la possibilita' di apportare delle modifiche ai valori precedentemente memorizzati oppure di cancellare il settore corrente. La modifica del nome di un settore si riflette nelle eventuali informazioni ad esso connesse. La cancellazione di un settore implica la cancellazione di tutte le informazioni ad esso logicamente connesse.

Se l'utente opera su un settore gia' esistente cancellandolo o aggiornandone il nome, il sistema ripropone il forma che permette di specificare il nome di un settore relativamente al progetto corrente.

Se l'utente inserisce un nuovo settore o aggiorna i dati, ma non il nome, relativi ad un settore gia' esistente, il sistema visualizza un MENU' che permette la definizione di oggetti nell'ambito del settore e del progetto corrente (vedi figura 9).

<<MENU'>>:

Tramite la scelta di una certa opzione del MENU' si ha la possibilita' di specificare il nome di uno dei seguenti oggetti : classe, proprieta', tipo, operazione, attivita'.

Tramite l'opzione 0 si puo' uscire dal MENU' facendo ritorno al forma che permette di definire il nome di un settore (vedi figura 45 e 46).

Il sistema realizza la connessione degli oggetti su cui si opera con il settore ed il progetto corrente.

Ogni opzione del MENU' attiva una sottoapplicazione che verifica l'esistenza dell'oggetto specificato. Se l'oggetto non e' presente ne viene permessa l'inserzione tramite il relativo form che visualizza gli attributi della metaclassa relativa all'oggetto specificato. Se l'oggetto specificato esiste si entra nel form che da' la possibilita' di modificarne il nome; dopodiche' il pannello successivo visualizza, in una forma analoga a quella utilizzata per l'inserzione, il contenuto della metaclassa relativa all'oggetto sul quale si opera (vedi esempio da figura 23 a figura 26).

A questo punto si ha la possibilita' di apportare delle modifiche ai valori precedentemente memorizzati oppure di cancellare l'oggetto corrente. La modifica del nome di un oggetto si riflette nelle eventuali informazioni ad esso connesse.

In caso di cancellazione di un elemento di una metaclassa, il sistema provvede, se necessario, ad eliminare le altre informazioni che risultano essere logicamente connesse.

Difettuata una qualsiasi transazione il sistema ripropone il MENU' eccetto che nei due casi qui segnalati:

- Dal momento che e' molto frequente operare sulle proprieta' di una classe si e' ritenuto opportuno, nel caso di inserimento o di modifica di quest'ultima, far passare il sistema direttamente alla sotto-applicazione che gestisce le proprieta' della classe (vedi esempio da figura 10 a figura 12, da figura 23 a figura 27).

- La descrizione del corpo delle attivita' e delle operazioni presenta caratteristiche che ne rendono di fatto impossibile la reindirizzazione come campi caratteri di lunghezza definita.

Per ovviare a questo problema, si e' ritenuto opportuno permettere all'utente di utilizzare l'editore di sistema, senz'altro a lui piu' familiare e di piu' agevole uso; di conseguenza nella specifica delle attivita' e delle operazioni e' previsto un campo in cui l'utente puo' digitare il nome del "corpo".

Se viene specificato un valore per il campo "corpo", l'utente si trova in 'EDIT' del file avente come nome il nome del corpo specificato.

Dietro richiesta del sistema, l'utente puo' decidere di attivare una funzione che provvede a strutturare la descrizione del corpo secondo uno standard di documentazione analogo a quello adottato nella pr-

grammazione strutturata (vedi esempio da figura 40 a figura 44).

5.2. AMBIENTE OPERATIVO.

La descrizione dell'ambiente operativo viene fatta solo in relazione al sistema CMS in quanto e' proprio sotto tale sistema che e' stato realizzato il prodotto e, viste le sue caratteristiche, si ritiene l'ambiente piu' idoneo di utilizzo.

Cio' non toglie che l'utilizzo sotto TSC o PC/FOCUS puo' essere considerato completamente trasparente al sistema.

Il sistema realizzato e' costituito da tre parti fondamentali: una parte procedurale, una parte descrittiva e una parte contenente i dati su cui opera la parte procedurale in relazione alla parte descrittiva.

La parte procedurale e' costituita da tutti i files di tipo FOCXEC che vengono richiamati uno con l'altro secondo la logica con cui e' stato realizzato il sistema.

La parte descrittiva e' costituita dalla descrizione dei files del sistema ognuno dei quali risiede su un file di tipo MASTEL.

Sia la parte procedurale che quella descrittiva non sono modificabili e sono quindi disponibili in sola lettura all'utente.

La parte dei dati, contenuta in files di tipo FOCUS, e' invece disponibile all'utente per la modifica.

Possiamo dunque configurare l'ambiente operativo come segue:

- un disco contenente il FOCUS accessibile in sola lettura;
- un disco contenente le procedure e le descrizioni dei dati accessibile in sola lettura;
- un disco contenente i dati disponibile alla modifica dell'utente.

Procedura di utilizzo:

All'utente e' richiesto di collegarsi ad una macchina virtuale, contenente il sistema, secondo le procedure di 'LOGON' classiche del CMS. Eseguita tale procedura si presuppone l'esistenza di un insieme di comandi che prepari l'ambiente operativo come sopra descritto; tale procedura terminera' con l'esecuzione di un comando che rendera' operativo il FOCUS.

A questo punto si configurano due situazioni temporalmente distinte:

1. L'utente utilizza per la prima volta il sistema.

In tal caso il sistema va inizializzato, nel senso che va inizializzato il supporto dati relativo alle 'Associazioni'. La procedura d'inizializzazione viene richiamata ed eseguita con il comando "EXECUTE SDD001". In risposta a tale comando il sistema richiede un nome di associazione fittizio.

Si consiglia di non inserire alcun valore premendo il tasto 'ENTER' a vuoto.

2. L'utente ha già proceduto ad inizializzare il sistema.

Immettendo da tastiera il comando "EXECUTE SDD" (semantic data design) viene visualizzato il primo pannello dell'applicazione.

Di seguito, in modo schematico, un esempio di sessione:

```
LOGON VMID PASS           { Si chiede l'inizializzazione di
                          una macchina virtuale di nome
                          VMID e di password PASS      )
FOCUS                     { Manda in esecuzione una procedura
                          che inizializza l'ambiente FOCUS)
EXECUTE SDB               { Viene attivato il sistema
                          realizzato                      )
/* ..... Utilizzo del sistema ..... */
FIN                       { Disconnette l'utente dal FOCUS )
LOGOFF                    { Disconnette l'utente dal VM/370 )
```

Nel prossimo paragrafo verrà descritto un esempio reale in modo tale da rendere più chiare le modalità di utilizzo del sistema realizzato.

5.3. ESEMPIO DI SESSIONE.

Viene presentata la descrizione di alcuni requisiti del sistema informativo per la gestione dei servizi del CNOCE.

L'esempio e' tratto da uno studio presentato in [Wacci,83] e [Saraglia,83]

Il CNUCF svolge sia attivita' di ricerca che erogazione di servizi; il nostro interesse e' rivolto al secondo tipo di attivita'. Il servizio CNUCF e' suddiviso in quattro principali aree di lavoro:

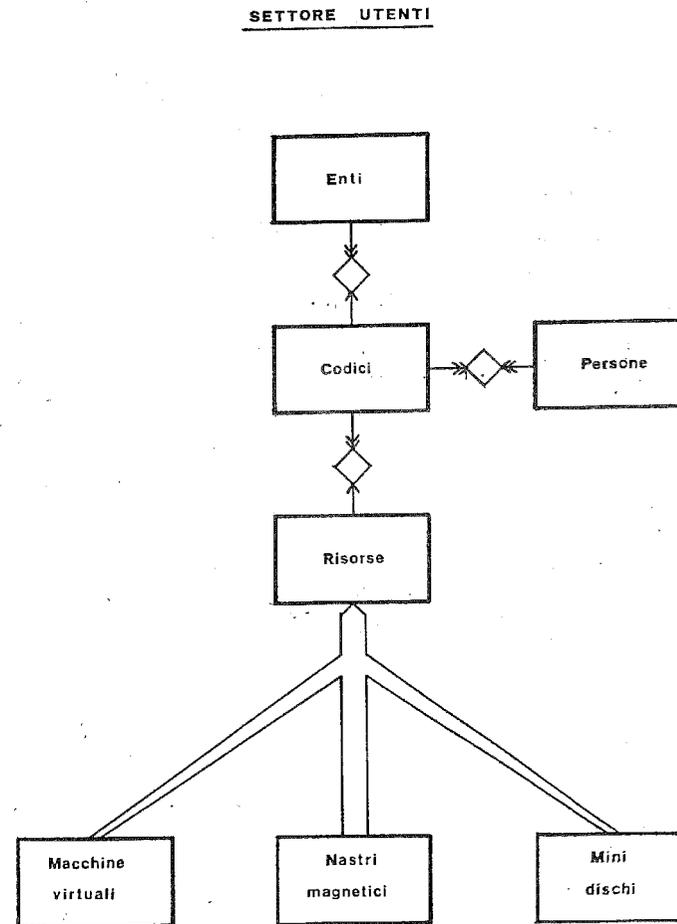
- area FP
- area utenti
- risorse del centro
- servizi interni

Abbiamo creduto sufficiente, per il nostro scopo, esaminare un unico ambiente e abbiamo scelto l'area utenti per le piu' vaste implicazioni che comporta.

Per la definizione dello schema concettuale sono state identificate le seguenti entita': Enti, Codici, Risorse, Persone.

Altre tre entita' sono state viste come specializzazioni della entita' Risorse. Esse sono: Macchine virtuali, Nastri magnetici, Mini dischi.

Nella seguente figura e' rappresentato lo schema concettuale del settore utenti.



Sono stati definiti solo alcuni requisiti in quanto lo scopo principale e' quello di evidenziare le caratteristiche del sistema realizzato.

Vengono mostrati di seguito i pannelli tramite i quali l'utente interagisce con il sistema.

Per non appesantire l'esempio di sessione, i pannelli vengono presentati con le definizioni gia' digitate.

I campi contrassegnati da un asterisco, visualizzati dal sistema, sono campi protetti.

La scritta racchiusa tra gli asterischi nella parte superiore di ogni pannello e' protetta. A partire dalla logica seguente il "MENU" viene visualizzata nella parte destra di tale scritta un'informazione relativa ad ogni pannello: "screen (L/S, n, s)":

- L o S e' seconda se siamo in un pannello relativo all'inserzione o all'aggiornamento di una certa informazione;
- n e' il numero che corrisponde all'opzione specificata al livello del "MENU"
- s e' il numero progressivo che si riferisce alla logica della particolare sott'applicazione.

```
G *****
                                INSERZIONE   PROGETTO
                                09.05.02   03/21/86
                                *****
```

NOME PROGETTO:

Per continuare =====> premere ENTER
PF01/PF03 =====> FOCUS

4A*

M 1 Host 3279 Session

Figura 1. Form iniziale: viene visualizzato conseguentemente all'invio del comando "EXECUTE SDD"; l'utente puo' specificare il campo come mostrato nella Figura 2 a pag. 91

6 *****

09.05.02 INSERZIONE PROGETTO
03/21/86

NOME PROGETTO: Gestione Utenza CNUCE

Per continuare =====> premere ENTER
PF01/PF03 =====> FOCUS

4A*

M 1 Host 3279 Session

Figura 2. Definizione del nome progetto: premendo
ENTER si ottiene il form di Figura 3 a pag.

92

6 *****

Inserzione del progetto Gestione Utanza CNUCE

DESCRIZIONE:

RESPONSABILE:

PASSWORD:

DATA INIZIO:

DATA ULTIMA MODIFICA:

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD/HELP ENTER=CONTINUE

4A M 1 Host 3279 Session

Figura 3. Inserzione del progetto 'Gestione Utanza CNUCE': volendo prendere visione delle caratteristiche dei campi che descrivono un progetto si preme PF08 ottenendo la Figura 4 a pag. 93

6 *****

HELP		PROGETTO	
Attributo	Tipo	Lunghezza	Descrizione
DESCRIZIONE	STRINGA	240	Descrizione in linguaggio naturale del progetto.
RESPONSABILE	STRINGA	30	Persona responsabile della evoluzione del progetto.
PASSWORD	STRINGA	10	Codice di riconoscimento della persona responsabile del progetto.

PF08 continua Help

4A*

M 1 Host 3279 Session

Figura 4. Help progetto: premendo PF08 si ottiene il form di Figura 5 a pag. 94

6

DATA INIZIO	INTERO	6	Data in cui ha inizio la stesura del progetto. Si specifica nell'ordine: giorno, mese, anno.
DATA ULTIMA MODIFICA	INTERO	6	Data dell'ultima modifica del progetto. Si specifica nell'ordine: giorno, mese, anno.

4A*

M 1 Host 3279 Session

Figura 5. Continua help progetto: tramite la PF07 si risale al pannello di Figura 3 a pag. 92 per definire il progetto come descritto in Figura 6 a pag. 95

6 *****

Inserzione del progetto Gestione Utente CNUCE

DESCRIZIONE: Vengono presentati i requisiti del sistema informativo
per la gestione dei servizi del CNUCE.

RESPONSABILE: O. Signore

PASSWORD: A18396

DATA INIZIO: 210386

DATA ULTIMA MODIFICA:

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD/HELP ENTER=CONTINUE

4A-

M 1 Host 3279 Session

Figura 6. Definizione del progetto: premendo ENTER si
ottiene il form di Figura 7 a pag. 96

6 *****

INSERZIONE SETTORE

NOME SETTORE: Area utenti

Per continuare =====> premere ENTER
^F01/PF03 =====> FOCUS

4A*

M 1 Host 3279 Session

Figura 7. Definizione del nome settore: premendo ENTER
si ottiene il form di Figura 8 a pag. 97

6 *****

Inserzione del settore Area utenti

*PROGETTO:Gestione Utenza CNUCE

DESCRIZIONE: Questo settore regola la dinamica della gestione delle risorse che il CNUCE mette a disposizione della propria utenza.

DATA ULTIMA MODIFICA:

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5=

7= BACK 8=FORWARD/HELP

6=

6=

ENTER=CONTINUE

4A.

M 1 Host 3279 Session

Figura 8. Definizione del settore 'Area utenti': premendo ENTER si passa al successivo form di Figura 9 a pag. 98 che visualizza il MENU'.

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 1

OPZIONE	FUNZIONE
1	Operare sulla classe: Enti
2	Operare sulle proprieta' di:
3	Operare sui tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

4A*

M 1 Host 3279 Session

Figura 9. Form relativo al MENU': l'utente puo' specificare una tra le opzioni elencate ed il nome dell'oggetto su cui desidera operare; premendo ENTER si ottiene il form di Figura 10 a pag. 99

6 *****

Inserzione della classe Enti screen I.1.0

*SETTORE:Area utenti

DESCRIZIONE: Questa classe descrive tutti gli enti che usufruiscono
del servizio CNUCE.

UTENTI:

SOTTOINSIEME DI:

PARTIZIONE DI:

RESTRIZIONE DI:

CLAUSOLA DI RESTRIZIONE:

PF8 per continuare l' immissione

4A*

M 1 Host 3279 Session

Figura 10. Definizione della classe 'Enti': premendo
PF08 si ottiene il form di Figura 11 a pag.
100 che permette di continuare l'immissione
dei dati relativi alla classe specificata.


```

6 *****
  ATRIBUTO O ASSOCIAZIONE                               screen I.2.0
    DELLA CLASSE Enti
  *****
  ATRIBUTO :

  O

  ASSOCIAZIONE : Enti-Codici
                CODOMINIO : Codici

  PFK
  1=FOCUS  2=UNDO  3=FOCUS  4=END  5=      6=
  7= BACK  8=FORWARD      ENTER=CONTINUE

```

4A*

M 1 Host 3279 Session

Figura 12. Definizione del nome di un'associazione per
 la classe corrente: premendo ENTER viene
 visualizzato il form di Figura 13 a pag. 102

6 *****

Associazione della Classe Enti screen I.2.2

*NOME ASSOCIAZIONE:Enti-Codici

*CODOMINIO:Codici

NOME ASSOCIAZIONE INVERSA: Codici-Enti

DESCRIZIONE: descrive l'associazione tra gli enti che hanno fatto
richiesta di accesso alle risorse di calcolo e i
codici assegnati.

MULTIPLA: si MODIFICABILE: si

TOTALE: si PROBABILITA' ESISTENZA(%):

CARDINALITA' MINIMA: 1 MASSIMA: MEDIA: 20

CRESCITA ANNUA(%):

VOLATILITA': I codici vengono cambiati con frequenza annuale.

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5=

7= BACK 8=FORWARD/HELP

6=
ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 13. Definizione della associazione
specificata: premendo ENTER viene creata
automaticamente l'associazione inversa spe-
cificata, dopodiche' si ottiene la Figura 14
a pag. 103

6

La classe
Codici

specificata come codominio non esiste

PF11 conferma inserzione
PF4 per annullare l' operazione

4A*

M 1 Host 3279 Session

Figura 14. Messaggio del sistema: premendo PF11 il sistema crea automaticamente la classe 'Codici', dopodiche' visualizza il form di
Figura 15 a pag. 104

```

6 *****
ATTRIBUTO O ASSOCIAZIONE                                screen I.2.0
  DELLA CLASSE Enti
*****
ATTRIBUTO :

O

ASSOCIAZIONE :
  CODOMINIO :

PFK
1=FOCUS  2=UNDO  3=FOCUS  4=END  5=      6=
7= BACK  8=FORWARD      ENTER=CONTINUE

```

4A*

M 1 Host 3279 Session

Figura 15. Possibilita' di definire una proprieta': decidendo di terminare l'immissione di proprieta' per la classe corrente si preme PF04; il sistema fa ritorno al MENU' di Figura 16 a pag. 105

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 1

OPZIONE	FUNZIONE
1	Operare sulla classe: Nastri magnetici
2	Operare sulle proprieta' di:
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

4A*

M 1 Host 3279 Session

Figura 16. Definizione del nome di una classe: pre-
mendo ENTER si ottiene il form di Figura 17
a pag. 106.

Inserzione della classe Nastri magnetici screen I.1.0

*SETTORE:Area utenti
DESCRIZIONE:

UTENTI:

SOTTOINSIEME DI: Risorse

PARTIZIONE DI: Risorse

RESTRIZIONE DI:
CLAUSOLA DI RESTRIZIONE:

PF8 per continuare l'immissione
Le tre modalita' devono essere mutuamente esclusive

4A*

M 1 Host 3279 Session

Figura 17. Definizione della classe 'Nastri magnetici': avendo erroneamente definito la classe sia come sottoinsieme che come partizione della classe 'Risorse', premendo ENTER il sistema rifiuta l'immissione visualizzando nuovamente lo stesso form con l'aggiunta del messaggio di errore.

```

) *****
Inserzione della classe Nastri magnetici                      screen I.1.0
) *****
*SETTORE:Area utenti
-DESCRIZIONE:
)

) UTENTI:

) SOTTOINSIEME DI: Risorse
) PARTIZIONE DI:
) RESTRIZIONE DI:
)   CLAUSOLA DI RESTRIZIONE:
)
) PF8 per continuare l'immissione
) Le tre modalita' devono essere mutuamente esclusive
)
) 4A. M 1 Host 3279 Session
)

```

Figura 18. Correzione errore: premendo ENTER si ottiene la Figura 19 a pag. 108

La classe
Risorse

specificata come classe generale non esiste

PF11 conferma inserzione
PF4 per annullare l' operazione

4A*

M 1 Host 3279 Session

Figura 19. Messaggio del sistema: premendo PF11 il sistema crea automaticamente la classe 'Risorse', dopodiche' visualizza il form di
Figura 20 a pag. 109

6 *****

ATTRIBUTO O ASSOCIAZIONE screen I.2.0
DELLA CLASSE Nastri magnetici

ATTRIBUTO : Posizione nastroteca

O

ASSOCIAZIONE :
CODOMINIO :

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5=
7= BACK 8=FORWARD ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 20. Definizione del nome di un attributo per la
classe corrente: premendo ENTER viene vi-
sualizzato il form di Figura 21 a pag. 110

6 *****

Attributo della Classe Nastri magnetici screen I.2.1

*NOME ATTRIBUTO : Posizione nastroteca

DESCRIZIONE: Descrive la posizione del nastro magnetico
nella nastroteca.

TIPO: String

LUNGHEZZA: 4 MISURA:

ISTANZE PREVISTE: 3000 DI CUI 3000 DIVERSE

OPZIONALE: no PROBABILITA' DI ESISTENZA(%):

DERIVATO: MODIFICABILE: si COMPOSTO: no

SEQUENZA: no MAX_IN_SEQ : MIN_IN_SEQ :

ASSERZIONE:

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5=

7= BACK 8=FORWARD/HELP

6=
ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 21. Definizione dello specificato
attributo: premendo ENTER si passa al suc-
cessivo form di Figura 22 a pag. 111 che da'
nuovamente la possibilita' di definire una
proprietà per la classe corrente.

6 *****

ATTRIBUTO O ASSOCIAZIONE screen I.2.0
DELLA CLASSE Nastri magnetici

ATTRIBUTO :

O

ASSOCIAZIONE :
CODOMINIO :

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5=
7= BACK 8=FORWARD 6=
ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 22. Possibilita' di definire una
proprietà: decidendo di terminare l'immissione di proprietà per la classe corrente si preme PF04; il sistema fa ritorno al MENU* di Figura 23 a pag. 112

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 1

OPZIONE	FUNZIONE
1	Operare sulla classe: Codici
2	Operare sulle proprieta' di:
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

4A-

M 1 Host 3279 Session

Figura 23. Definizione del nome di una classe: detta classe e' stata creata automaticamente dal sistema (vedi figura 14). Premendo ENTER si ottiene il form di Figura 24 a pag. 113

6 *****
MODIFICA DELLA CLASSE: Codici

NUOVO NOME CLASSE : Codici

Eventualmente modificare il nome classe. ENTER per continuare

4A*

M 1 Host 3279 Session

Figura 24. Possibilita' di modifica del nome della
classe specificata : premendo ENTER si ot-
tiene il form di Figura 25 a pag. 114

6 *****

Modifica della classe Codici screen U.1.0

*SETTORE:Area utenti
*NUOVO NOME:Codici
DESCRIZIONE:

UTENTI:

SOTTOINSIEME DI:

PARTIZIONE DI:

RESTRIZIONE DI:
CLAUSOLA DI RESTRIZIONE:

PF8 per continuare l'immissione

4A*

M 1 Host 3279 Session

Figura 25. Modifica della classe 'Codici': premendo PF08 si ottiene il form di Figura 26 a pag. 115 che permette di continuare l'aggiornamento dei dati relativi alla classe specificata.

5 *****

ATTRIBUTO O ASSOCIAZIONE screen I.2.0
DELLA CLASSE Codici

ATTRIBUTO :

O

ASSOCIAZIONE :
CODOMINIO :

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 27. Possibilita' di definire una proprieta' per
la classe corrente: scegliendo di terminare
si preme PF04 facendo ritorno al MENU' di
Figura 28 a pag. 117

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 2

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di: Nastri
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

Errore..la classe specificata non esiste!

Figura 28. Scelta di operare sulle proprieta' di una classe: avendo specificato una classe non esistente, premendo ENTER, il sistema rifiuta l'immissione visualizzando un messaggio errore.

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 2

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di: Nastri magnetici
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

Errore..la classe specificata non esiste!

Figura 29. Correzione errore: premendo ENTER si ottiene la Figura 30 a pag. 119

6 *****

ATTRIEUTO O ASSOCIAZIONE screen I.2.0
DELLA CLASSE Nastri magnetici

ATTRIBUTO : Data di eliminazione

O

ASSOCIAZIONE :
CODOMINIO :

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 30. Definizione del nome di un attributo per la
classe corrente: premendo ENTER viene vi-
sualizzato il form di Figura 31 a pag. 120

6 *****

Attributo della Classe Nastri magnetici screen I.2.1

*NOME ATTRIBUTO :Data di eliminazione

DESCRIZIONE: Data in cui il supporto viene eliminato
dalla nastroteca del CNUCE.

TIPO: UnaData

LUNGHEZZA: MISURA:

ISTANZE PREVISTE: DI CUI DIVERSE

OPZIONALE: PROBABILITA' DI ESISTENZA(%):

DERIVATO: MODIFICABILE: COMPOSTO:

SEQUENZA: MAX_IN_SEQ : MIN_IN_SEQ :

ASSERZIONE:

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5=

7= BACK 8=FORWARD/HELP

6=
ENTER=CONTINUE

4A*

M 1 Host 3279 Session

Figura 31. Definizione dello specificato attributo: premendo ENTER si passa al successivo form di Figura 32 a pag. 121 che da' nuovamente la possibilita' di definire una proprieta' per la classe corrente.

```

*****
C *****
ATTRIBUTO O ASSOCIAZIONE                                screen I.2.0
          DELLA CLASSE Nastri magnetici
*****
ATTRIBUTO :

O

ASSOCIAZIONE :
          CODOMINIO :

PFK
1=FOCUS  2=UNDO  3=FOCUS  4=END  5=          6=
7= BACK  8=FORWARD          ENTER=CONTINUE

```

4A*

M 1 Host 3279 Session

Figura 32. Possibilita' di definire una proprieta': decidendo di terminare l'immissione di proprieta' per la classe corrente si preme PF04; il sistema fa ritorno al MENU' di Figura 33 a pag. 122

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 3

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di:
3	Operare sul tipo: UnaData
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

Figura 33. Definizione del nome di un tipo: premendo
ENTER si ottiene il form di Figura 34 a pag.

123

6 *****

Inserzione del tipo UnaData screen I.3.0

DESCRIZIONE: Tipo che modella una data.

Esso e' costituito dai campi Giorno, Mese e Anno.

ASSERZIONE: Il Giorno e' compreso nell'intervallo (1,31), il mese
nell'intervallo (1,12), l'Anno e' > 1985.

DATA ULTIMA MODIFICA:

PFK

1=FOCUS 2=UNDO 3=FOCUS 4=END 5=

7= BACK 8=FORWARD/HELP

6=

ENTER=CONTINUE

Figura 34. Definizione del tipo 'UnaData': premendo

ENTER il sistema fa ritorno al MENU' di Fi-

gura 35 a pag. 124

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 4

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di:
3	Operare sul tipo:
4	Operare sull' operazione: Lista codici CNUCE attivi
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

4A*

M 1 Host 3279 Session

Figura 35. Definizione del nome di un'operazione: pre-
mendo ENTER si ottiene il form di Figura 36
a pag. 125

6 *****

Inserzione della operazione Lista codici CNUCE attivi screen I.5.0

*SETTORE:Area utenti

DESCRIZIONE: Si individuano i codici CNUCE con la voce data scadenza
codice non impostata o, se impostata, superiore alla
data relativa al giorno in cui viene fatta eseguire
l'operazione.

FREQUENZA DI UTILIZZO: 1 PERIODO: mese

PRECONDIZIONI: Deve esistere una data di scadenza codice non impostata
o, se impostata, deve essere maggiore della data
corrente.

PF8 per continuare l'immissione

Figura 36. Definizione dell'operazione 'Lista codici
CNUCE attivi': premendo PF08 si ottiene il
form di Figura 37 a pag. 126 che permette di
continuare l'immissione dei dati relativi
all'operazione specificata.

6 *****

Continua inserzione della operazione Lista codici CNUCE attivi

POSTCONDIZIONI:

ECCEZIONI:

CORPO:

DATA ULTIMA MODIFICA:

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD/HELP ENTER=CONTINUE

Figura 37. Continua la definizione dell'operazione
'Lista codici CNUCE attivi' : premendo
ENTER il sistema fa ritorno al MENU' di Fi-
gura 38 a pag. 127

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 5

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di:
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita': Evasione richieste di utenti
0	EXIT

Per continuare =====> premere ENTER

4A.

M 1 Host 3279 Session

Figura 38. Definizione del nome di un'attivita': pre-
mendo ENTER si ottiene il form di Figura 39
a pag. 128

6 *****

Inserzione dell' attivita' Evasione richieste di utenti screen I.4.0

*SETTORE:Area utenti
UTENTI: Utenti CNUCE

DESCRIZIONE:

FREQUENZA DI UTILIZZO: 10 PERIODO: giorno
PRIORITA'(alta/bassa/media): alta
LIVELLO(strategico/gestionale/operativo): gestionale
PRECONDIZIONI:

PF8 per continuare l' immissione

4A.

M 1 Host 3279 Session

Figura 39. Definizione dell'attivita' 'Evasione richieste di utenti': premendo PF08 si ottiene il form di Figura 40 a pag. 129 che permette di continuare l'immissione dei dati relativi all'attivita' specificata.

6 *****

Continua inserzione della attivita' Evasione richieste di utenti

POSTCONDIZIONI:

ECCEZIONI:

PROCESSO: Evasione
MODALITA'(batch/online): online
TEMPO DI ATTESA MEDIO: MASSIMO:
DATA ULTIMA MODIFICA:

PFK
1=FOCUS 2=UNDO 3=FOCUS 4=END 5= 6=
7= BACK 8=FORWARD/HELP ENTER=CONTINUE

Figura 40. Continua la definizione dell'attivita' 'Evasione richieste di utenti' : premendo ENTER il sistema entra in 'Edit' di un file avente il nome del corpo specificato nell'attivita'. Vedi Figura 41 a pag. 130

```
6 Evasione PROC      Al  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0
```

```
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
00000 * * * Top of File * * *  
00001 * * * End of File * * *
```

Figura 41. File 'Evasione': effettuando un'operazione di input si descrive il corpo dell'attivit  come in Figura 42 a pag. 131

6 Evasione PROC A1 F 80 Trunc=80 Size=6 Line=0 Col=1 Alt=12

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
00000 * * * Top of File * * *
00001 /* Quando un rappresentante richiede di autorizzare un presentatore */
00002 IF ( (codice CNUCE="attivo") AND (codice CNUCE appartiene
00003                                     al rappresentante) )
00004         THEN inserisci dati presentatore
00005     ELSE rifiuta richiesta
00006         ENDIF
00007 * * * End of File * * *
```

====> file
4A*

M 1 Host 3279 Session

Figura 42. Descrizione del corpo dell'attivita' 'Eva-
sione richieste di utenti': tramite il co-
mando 'File' si ottiene la memorizzazione di
tale informazione, dopodiche' il sistema in-
via il messaggio di Figura 43 a pag. 132

6VUOI LA FORMATTAZIONE DELLA PROCEDURA?

RISPONDI SI O NO

SI

4A*

RUNNING ICNUCEVM
M 1 Host 3279 Session

Figura 43. Messaggio del sistema: rispondendo 'si' il sistema visualizza la descrizione, opportunamente formattata, del corpo dell'attività come in Figura 44 a pag. 133 ; dopodiché fa ritorno al MENU' di Figura 45 a pag. 134

```
/* Quando un rappresentante richiede di autorizzare un presentatore */  
IF ( (codice CNUCE="attivo") AND (codice CNUCE appartiene  
    al rappresentante) )  
THEN inserisci dati presentatore  
ELSE rifiuta richiesta  
ENDIF
```

====>
4A*

M 1 Host 3279 Session

Figura 44. Formattazione del corpo dell'attività.

6 *****

MENU PER LA DEFINIZIONE DI OGGETTI

OPZIONE DESIDERATA: 0

OPZIONE	FUNZIONE
1	Operare sulla classe:
2	Operare sulle proprieta' di:
3	Operare sul tipo:
4	Operare sull' operazione:
5	Operare sull' attivita':
0	EXIT

Per continuare =====> premere ENTER

Figura 45. MENU': premendo ENTER il sistema fa ritorno
al form di Figura 46 a pag. 135

```
*****  
; *****  
                INSERZIONE      SETTORE  
*****
```

NOME SETTORE:

Per continuare =====> premere ENTER
PF01/PF03 =====> FOCUS

Figura 46. Possibilita' di specificare il nome di un settore relativamente al progetto corrente : decidendo di terminare la sessione si preme PF01 (o equivalentemente PF03) facendo cosi' ritorno all'ambiente FOCUS.

Rapporto relativo al Progetto Gestione Utenza CNUCI

RESPONSABILE C. Signore

DESCRIZIONE Vengono presentati i requisiti del sistema informativo
per la gestione dei servizi del CNUCI.

DATA INIZIO PROGETTO 24/3/86

DATA ULTIMA MODIFICA

Rapporto relativo al Settore Area utenti del Progetto Gestione Utenza CNUCE

DESCRIZIONE Questo settore regola la dinamica della gestione delle risorse che il CNUCE mette a disposizione della propria utenza.

DATA ULTIMA MODIFICA

Rapporto concernente i settori definiti
nel Progetto Gestione Utente CNUCE

LIST

- 1 Area utenti
Questo settore regola la dinamica della gestione delle risorse che il CNUCE mette a disposizione della propria utenza.

DATA ULTIMA MODIFICA

- 2 Area TP
Questo settore regola l'accesso alle risorse di calcolo del CNUCE attraverso una rete di collegamenti TP.

DATA ULTIMA MODIFICA

- 3 Risorse del centro

DATA ULTIMA MODIFICA

- 4 Servizi interri

DATA ULTIMA MODIFICA

Reporto concernente le attivita' definite
nel Settore Area utenti del Progetto Gestione Utenza (NUCE)

LIST

- 1 Aggiornamento
Questa attivita' riguarda variazioni di aggiornamento
di dati gia' esistenti.

DATA ULTIMA MODIFICA

- 2 Evasione richieste di utenti

DATA ULTIMA MODIFICA

- 3 Evasione richieste di utenza
Questa attivita' descrive tutto cio' che comporta la
richiesta da parte di un ente di diventare utente
del CNCCI.

DATA ULTIMA MODIFICA

- 4 Produzione di documenti
Questa attivita' comprende sia comunicazioni che il CNUCE
invia ai suoi utenti in seguito all'accettazione di una
qualsiasi richiesta, sia liste utili alla gestione del
servizio.

DATA ULTIMA MODIFICA

1 PAGE 1

Rapporto relativo alla Attivita' Evasione richieste di utenti definita
nel Settore Area utenti del Progetto Gestione Utenza CNUCE

DESCRIZIONE

FREQUENZA DI DURATEC	10
PERIODO	giornC
UTENTI	Utenti CNUCE

PRIORITA	alta
MODALITA	online
LIVELLO	gestionale
PRECONDIZIONI	

POSTCONDIZIONI

ECCEZIONI

ECCESSO	Evasione
TEMPO DI ATTESA MEDIO	
TEMPO DI ATTESA MASSIMO	
DATA ULTIMA MODIFICA	

Lista delle classi definite
nel Settore Area utenti del Progetto Gestione Utenza CRUCI

LIST

1 Codici

2 Enti

3 Macchine virtuali

4 Mini dischi

5 Nastri magnetici

6 Persone

7 Risorse

Rapporto concernente gli attributi definiti
nella Classe Nastri magnetici

LIST

- 1 Data di eliminazione
Data in cui il supporto viene eliminato
dalla nastroteca del CNUCL.

- 2 Data di riserva
Data in cui e' stato inserito il nastro magnetico
nella nastroteca del CNUCL.

- 3 Data ultimo addebito
Data in cui e' stata emessa l'ultima fattura
di addebito all'utente.

- 4 Dimensione supporto

- 5 Identificazione supporto
L'identificazione, fornita dall'utente, corrisponde
al nome del supporto.

- 6 Password

- 7 Posizione nastroteca
Descrive la posizione del nastro magnetico
nella nastroteca.
-

1 PAGE 1

Rapporto relativo alla Operazione Lista codici CNUCE attivi definita nel Settore Area utenti del Progetto Gestioneutenza CNUCE

DESCRIZIONE	Si individuano i codici CNUCE con la voce data scadenza codice non impostata o, se impostata, superiore alla data relativa al giorno in cui viene fatta eseguire l'operazione.
FREQUENZA DI UTILIZZO	1
PERIODO	mese
PRECONDIZIONI	Deve esistere una data di scadenza codice non impostata o, se impostata, deve essere maggiore della data corrente.
POSTCONDIZIONI	
ECCEZIONI	
CONFO	
DATA ULTIMA MODIFICA	

1 PAGE 1

Rapporto relativo alla classe Macchine virtuali del settore Area utenti
del progetto Gestione Utenza CNUCE

DESCRIZIONE

UTENTI

CARDINALITA MINIMA	
MASSIMA	1500
MEZIA	
CRESCENTE	
SOTTOINSIEME DI	Risorse
PARTIZIONE DI	
RESTRIZIONE DI	
CLAUSOLA DI RESTRIZIONE	
ATTRIBUTI CHIAVE	Nome macchina virtuale

ASSEZIONI

DATA ULTIMA MODIFICA

Rapporto relativo alla Associazione Enti-Codici avente
per dominio la Classe Enti, definita
nel Settore Area utenti del Progetto Gestione Utenza (NUCE)

NUCE ASSOCIAZIONE INVERSA	Codici-Enti
CODOMINIO	Codici
DESCRIZIONE	descrive l'associazione tra gli enti che hanno fatto richiesta di accesso alle risorse di calcolo e i codici assegnati.
TOTALITA	si
MOLTEPLICITA	si
MODIFICABILE	si
PROBABILITA DI ESISTENZA	
CARDINALITA MINIMA	1
CARDINALITA MASSIMA	
MEDIA	20
CRESCITA	
VOLATILITA	I codici vengono cambiati con frequenza annuale.

1 PAGE 1

Rapporto relativo al Tipo 00Indirizzo

DESCRIZIONE Modella un indirizzo. Esso e' costituito dai campi Via,
Numero Civico, Localita', CAP, Comune, Provincia.
La Via puo' essere: una Via, una Piazza, un Corso e un
Viale.

ASSENZIONI Il Numero Civico e' > 0.
Il CAP e' compreso nell'intervallo (0,100000).

DATA ULTIMA MODIFICA

5.5. I CONTROLLI REALIZZATI.

Il sistema esegue diversi controlli sui valori che l'utente fornisce come parametri delle definizioni; controlli necessari al mantenimento di uno stato consistente della metabase. Quando si verifica una condizione di errore il sistema visualizza un opportuno messaggio. Vengono ora elencati i controlli implementati:

Progetti:

- Il nome di un progetto non puo' essere la stringa vuota.
- Non e' possibile definire due progetti con lo stesso nome.

Settori:

- Il nome di un settore non può essere la stringa vuota.
- Ogni settore appartiene ad un progetto.
- I nomi dei settori appartenenti ad un progetto sono diversi.

Classi:

- Il nome di una classe non puo' essere la stringa vuota.
- Ogni classe appartiene ad un settore.
- I nomi delle classi appartenenti ad un settore sono diversi.
- Le icc gerarchie di generalizzazione devono essere mutuamente esclusive.
- La classe antecedente di una sottoclasse e' definita nello stesso settore della sottoclasse.
- Il valore medio della cardinalita' di una classe e' compreso tra il valore minimo e quello massimo.

Attributi:

- Il nome di un attributo non puo' essere la stringa vuota.
- Ogni attributo appartiene ad una classe.
- I nomi degli attributi appartenenti ad una classe sono diversi.
- La specifica della probabilita' di esistenza deve essere fatta in percentuale.

Associazioni:

- Il nome di un'associazione non puo' essere la stringa vuota.
- Per ogni associazione deve essere specificata la classe dominio.
- Per ogni associazione deve essere specificata la classe codominio.

- I nomi delle associazioni che pongano in relazione uno stesso dominio e codominio sono diversi tra loro.
- La classe dominio e quella codominio di una associazione sono definite nel settore corrente.
- La specifica della diretta e dell'inversa di una associazione e' consistente. Nella classe codominio e' definita un'associazione con il nome uguale a quello dell'inversa specificata nella classe corrente.
- Per ogni associazione deve essere specificato il nome dell'inversa.
- La specifica della probabilita' di esistenza deve essere fatta in percentuale.

Tipi:

- Il nome di un tipo non puo' essere la stringa vuota.
- I nomi dei tipi sono diversi.

Operazioni:

- Il nome di un'operazione non può essere la stringa vuota.
- Ogni operazione appartiene ad un settore.
- I nomi delle operazioni appartenenti ad un settore sono diversi.

Attività:

- Il nome di un'attività non può essere la stringa vuota.
- Ogni attività appartiene ad un settore.
- I nomi delle attività appartenenti ad un settore sono diversi.

6.0 CONCLUSIONI.

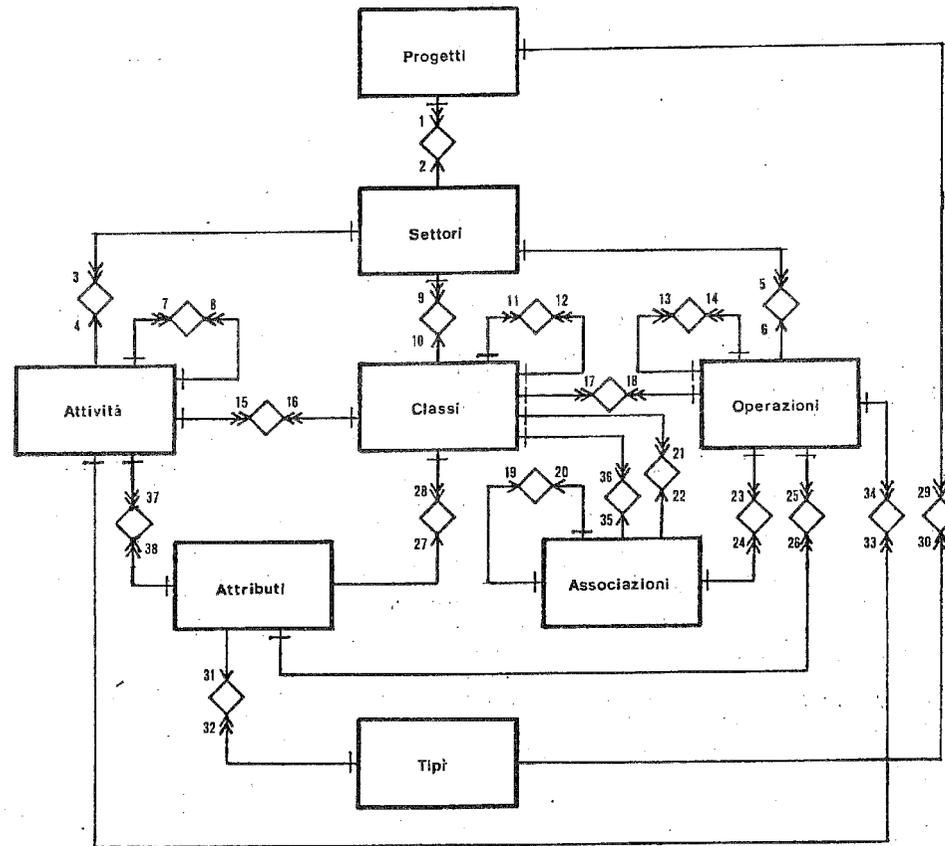
Cio' che ha motivato l'argomento della tesi e' stata la necessita' di formalizzare e strutturare la raccolta dei requisiti, tenendo conto delle diversita' delle categorie di persone interessate a questa fase.

Gli aspetti rilevanti della realizzazione sono :

- l'utilizzo di meccanismi di astrazione dei modelli semantici dei dati, previsti per la fase di progettazione concettuale;
- la specifica incrementale ed interattiva dei requisiti mediante l'utilizzo di forme;
- la memorizzazione dei requisiti in una base di dati;
- la realizzazione immediata dei controlli, nel senso che viene effettuata prima di modificare lo stato della base di dati;
- una facile consultazione dei requisiti inseriti;
- la scelta dell'ambiente hardware/software: il supporto realizzato e' funzionante sotto diversi Siste-

ai Operativi in ambiente Mainframe, ed e' possibile
con piccole modifiche renderlo disponibile su
Personal Computer.

APPENDICE A. SCHEMA CONCETTUALE DELLA METABASE.



1. Settori_definiti
2. Progetto_di_appartenenza
3. Attivita'_definite
4. Settore_di_definizione
5. Operazioni_definite
6. Settore_di_definizione
7. Attivita'_avviate
8. Attivita'_avvianti
9. Classi_definite
10. Settore_di_definizione
11. Sottoclassi
12. Classigenerali
13. Operazioni_invocate
14. Operazioni_chiaranti
15. Classi_referenziate
16. Attivita'_utilizzatrici
17. Operazioni_utilizzatrici
18. Classi_utilizzate
19. Inversa
20. Inversa
21. Associazione_dominio
- 22.ominio
23. Associazioni_utilizzate
24. Operazioni_utilizzanti
25. Attributi_utilizzati
26. Operazioni_utilizzanti
27. Classe_di_appartenenza

28. Attributi_della_classe
29. Tipi_definiti
30. Progetto_di_definizione
31. Tipo_attributo
32. Attributi_tipizzati
33. Operazioni_impiegate
34. Attivita'_utilizzatrici
35. Codominio
36. Associazioni_codominio
37. Attributi_della_attivita'
38. attivita'_utilizzanti

APPENDICE B. SCHEMA LOGICO E FISICO DELLA METABASE.

Per semplicità implementativa, e per ovviare ad alcuni limiti imposti dal FOCUS e dal suo modello dei dati (non modificabilità dei campi chiave, limitata possibilità di navigazione, impossibilità di rappresentare relazioni n : m) si è preferito realizzare la metabase di dati mediante otto distinti database FOCUS.

La consistenza delle informazioni è assicurata dalle procedure. In fase di produzione dei rapporti, i database vengono uniti dinamicamente mediante il comando JOIN.

B.1. PROGETTO

```
FILENAME=SDO,SUFFIX=PCC
SEGNAME=PROGETTO, SECURITY=SO
FIELDNAME=PROGETTO,          FDAO10,A10,FIELDTYPE=I,  $
FIELDNAME=DESCP1,           FDAO20,A60,                $
FIELDNAME=DESCP2,           FDAO21,A60,                $
FIELDNAME=DESCP3,           FDAO22,A60,                $
FIELDNAME=DESCP4,           FDAO23,A60,                $
FIELDNAME=RESPONSABILE,     FDAO30,A30,                $
FIELDNAME=PASSWORD,        FDAO40,A10,                $
FIELDNAME=DATA_INIZIO,      FDAO50,I6DNY,              $
FIELDNAME=DATA_FINE,        FDAO60,I6DNY,              $
```

E.2 SETTORE

```
FILENAME=SDDD, SUFFIX=IOC  
SEGNAME=SETTORE, SEGTYPE=SO  
FIELDNAME=SETTORE,          FCA010,A30, FIELDTYPE=I, $  
FIELDNAME=PROGETTON,       FCA000,A30, FIELDTYPE=I, $  
FIELDNAME=DESCS1,          FCA020,A60,          $  
FIELDNAME=DESCS2,          FCA021,A60,          $  
FIELDNAME=DESCS3,          FCA022,A60,          $  
FIELDNAME=DESCS4,          FCA023,A60,          $  
FIELDNAME=DATA_OLY_HOD,    FCA030,IS DMY,    $
```

B.3 CLASSE

```

FILENAME=GDD1, SUFFIX=FCC
PROGRAM=CLASSE, SECTYP=SO
FIELDNAME=NCMMCLASSE, F1A010,A30, FIELDTYPE=I, $
FIELDNAME=SPITOREC, F1A000,A30, FIELDTYPE=I, $
FIELDNAME=PROGETTCC, F1A000,A30, FIELDTYPE=I, $
FIELDNAME=UTENTIC1, F1A005,A30, $
FIELDNAME=UTENTIC2, F1A006,A30, $
FIELDNAME=UTENTIC3, F1A007,A30, $
FIELDNAME=CARDC.MIN, F1A015,I6, $
FIELDNAME=CARDC.MAX, F1A016,I6, $
FIELDNAME=CARDC.MED, F1A017,I6, $
FIELDNAME=CARDC.CHE, F1A019,I3, $
FIELDNAME=DESCC1, F1A020,A60, $
FIELDNAME=DESCC2, F1A021,A60, $
FIELDNAME=DESCC3, F1A022,A60, $
FIELDNAME=DESCC4, F1A023,A60, $
FIELDNAME=SCUICONSERIE, F1A030,A30, $
FIELDNAME=PARTIZIONE, F1A040,A30, $
FIELDNAME=RESISTIONE, F1A060,A30, $
FIELDNAME=CLA.RESTRIZ, F1A070,A30, $
FIELDNAME=CHIAVE1, F1A080,A30, $
FIELDNAME=CHIAVE2, F1A081,A30, $
FIELDNAME=ASSERZIONEC1, F1A090,A60, $
FIELDNAME=ASSERZIONEC2, F1A081,A60, $
FIELDNAME=DATA_UIG_MOD, F1A100,I6DM, $

```

E.4. ATTRIBUTO

```
FILENAME=SDD2, SUFFIX=PGC
SEGNAME=ATTRIB, SEGTYPE=SO
FIELDNAME=NOMEATTR, F2A010,A30, FIELDTYPE=I, $
FIELDNAME=ATTRIBUTO_DI, F2A060,A30, FIELDTYPE=I, $
FIELDNAME=SETTOGNEATTR, F2A080,A30, FIELDTYPE=I, $
FIELDNAME=PROGETTOATTR, F2A090,A30, FIELDTYPE=I, $
FIELDNAME=DESCR1, F2A020,A60, $
FIELDNAME=DESCR2, F2A021,A60, $
FIELDNAME=DESCR3, F2A022,A60, $
FIELDNAME=DESCR4, F2A023,A60, $
FIELDNAME=TIPOCORR, F2A015,A30, $
FIELDNAME=LUNGHESZA, F2A050,I6, $
FIELDNAME=MESURA, F2A040,A15, $
FIELDNAME=ISE.PREV, F2A050,I6, $
FIELDNAME=IST.DIV, F2A055,I6, $
FIELDNAME=DERIVATA, F2A060,A2, $
FIELDNAME=CONDIZIONALE, F2A070,A2, $
FIELDNAME=PROBABILITA, F2A075,I3, $
FIELDNAME=MODIFICABILE, F2A080,A2, $
FIELDNAME=COMPOSIZIA, F2A090,A2, $
FIELDNAME=SEQUENZA, F2A100,A2, $
FIELDNAME=MAXINSEI, F2A101,I6, $
FIELDNAME=MININSEI, F2A102,I6, $
FIELDNAME=ASSERZP1, F2A110,A60, $
FIELDNAME=ASSERZP2, F2A111,A60, $
```

B.5 ASSOCIAZIONE

```
FILENAME=SCD3, SUFFIX=FOC
SEGNAME=ASSOCIAZ, SEGTYPE=SO
  FIELDNAME=NCNEASSOC, F3A010, A30, FIELDTYPE=I, $
  FIELDNAME=DOMENIC, F3A000, A30, FIELDTYPE=I, $
  FIELDNAME=NCNECODON, F3A015, A30, FIELDTYPE=I, $
  FIELDNAME=SETTOREASS, F3ASEP, A30, FIELDTYPE=I, $
  FIELDNAME=PROGETTOASS, F3APRO, A30, FIELDTYPE=I, $
  FIELDNAME=DESCA1, F3A020, A60, $
  FIELDNAME=DESCA2, F3A021, A60, $
  FIELDNAME=DESCA3, F3A022, A60, $
  FIELDNAME=DESCA4, F3A023, A60, $
  FIELDNAME=ASS.TOTALE, F3A030, A2, $
  FIELDNAME=ASS.MULTIPL, F3A040, A2, $
  FIELDNAME=MODIFICABILE, F3A050, A2, $
  FIELDNAME=PROESISTE, F3A070, I3, $
  FIELDNAME=CARDA.MIN, F3A080, I6, $
  FIELDNAME=CARDA.MAX, F3A081, I6, $
  FIELDNAME=CARDA.MED, F3A082, I6, $
  FIELDNAME=CARDA.CRE, F3A083, I3, $
  FIELDNAME=VOLATTILTA, F3A100, A60, $
  FIELDNAME=NCME.ASS.INV, F3A110, A30, $
```

E.6 TIPO

```
FILENAME=SECD4, SUFFIX=FOC
SEGNAME=TIPO, SEGYEB=SO
FIELDNAME=MGNETIPO,          F4A010,A30, FIELDTYPE=I, $
FIELDNAME=PROGNETICI,       F4A010,A30, FIELDTYPE=I, $
FIELDNAME=DESC1,           F4A020,A60,          $
FIELDNAME=DESC12,          F4A021,A60,          $
FIELDNAME=DESC3,           F4A022,A60,          $
FIELDNAME=DESC4,           F4A023,A60,          $
FIELDNAME=ASSERZ1,          F4A030,A60,          $
FIELDNAME=ASSERZ2,          F4A031,A60,          $
FIELDNAME=ASSERZ3,          F4A032,A60,          $
FIELDNAME=DATA_ULT_MOD,     F4A040,I60MY,        $
```

E.7. APPENDICE*

```

FILENAME=SDDS,SUFFIX=FOC
SDGNAM=ALTEWITA,SECTYPE=SO
  FIELDNAME=NONFA,          F5A010,A30, FIELDTYPE=I,  $
  FIELDNAME=SETTOREA,      F5A000,A30, FIELDTYPE=I,  $
  FIELDNAME=PROGETTOA,     F5APRO,A30, FIELDTYPE=I,  $
  FIELDNAME=UTENNIA1,      F5A015,A30,          $
  FIELDNAME=UTENNIA2,      F5A016,A30,          $
  FIELDNAME=UTENNIA3,      F5A017,A30,          $
  FIELDNAME=DESCAT1,       F5A020,A60,          $
  FIELDNAME=DESCAT2,       F5A021,A60,          $
  FIELDNAME=DESCAT3,       F5A022,A60,          $
  FIELDNAME=DESCAT4,       F5A023,A60,          $
  FIELDNAME=FFBQUENZA,     F5A030,I6,           $
  FIELDNAME=PERIODC,       F5A031,A10,          $
  FIELDNAME=PRIORITA,      F5A080,A5,           $
  FIELDNAME=SCDALTITA,     F5A090,A7,           $
  FIELDNAME=LIVELLO,       F5A100,A10,          $
  FIELDNAME=PREC1,         F5A110,A60,          $
  FIELDNAME=PREC2,         F5A111,A60,          $
  FIELDNAME=PREC3,         F5A112,A60,          $
  FIELDNAME=PREC4,         F5A113,A60,          $
  FIELDNAME=POST1,         F5A120,A60,          $
  FIELDNAME=POST2,         F5A121,A60,          $
  FIELDNAME=POST3,         F5A122,A60,          $
  FIELDNAME=POST4,         F5A123,A60,          $
  FIELDNAME=ECCE21,        F5A130,A60,          $
  FIELDNAME=ECCE22,        F5A131,A60,          $
  FIELDNAME=ECCE23,        F5A132,A60,          $
  FIELDNAME=ECCE24,        F5A133,A60,          $
  FIELDNAME=PCCCSSC,       F5A140,A3,           $
  FIELDNAME=TEMPO_MESG,    F5A150,A10,          $
  FIELDNAME=TEMPO_MAY,     F5A151,A10,          $
  FIELDNAME=CATA_ULT_MDL,  F5A160,I6DM7,       $

```

E.8 OPERAZIONI

```
FILENAME=SDD6, SUFFIX=FOC
SIGNAME=OPERAZIO, SEQTYPE=SO
FIELDNAME=NCHBOP, F6A010,A30, FIELDTYPE=I, $
FIELDNAME=SETFORDC, F6A000,A30, FIELDTYPE=I, $
FIELDNAME=PEGETECO, F6APRO,A30, FIELDTYPE=I, $
FIELDNAME=DESCO1, F6A020,A60, $
FIELDNAME=DESCO2, F6A021,A60, $
FIELDNAME=DESCO3, F6A022,A60, $
FIELDNAME=DESCO4, F6A023,A60, $
FIELDNAME=PEQUT, F6A030,I6, $
FIELDNAME=PERIODC, F6A031,A10, $
FIELDNAME=PRECOND1, F6A110,A60, $
FIELDNAME=PRECOND2, F6A111,A60, $
FIELDNAME=PRECOND3, F6A112,A60, $
FIELDNAME=PRECOND4, F6A113,A60, $
FIELDNAME=POSTCOND1, F6A120,A60, $
FIELDNAME=POSTCOND2, F6A121,A60, $
FIELDNAME=POSTCOND3, F6A122,A60, $
FIELDNAME=POSTCOND4, F6A123,A60, $
FIELDNAME=ECCEZIONI1, F6A130,A60, $
FIELDNAME=ECCEZIONI2, F6A131,A60, $
FIELDNAME=ECCEZIONI3, F6A132,A60, $
FIELDNAME=ECCEZIONI4, F6A133,A60, $
FIELDNAME=CORPO, F6A140,A8, $
FIELDNAME=DATA_ULT_MCD, F6A150,I6DMY, $
```

APPENDICE C. CONTROLLI REALIZZATI E LORO GESTIONE.

In questa appendice viene presentato un elenco dei controlli che sono stati realizzati sulle definizioni dei requisiti.

C.1 CONTROLLI SUI PROGETTI:

Descrizione: il nome di un progetto non può essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un progetto.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome del progetto il sistema segnala con un messaggio tale inconsistenza.

Descrizione: non e' possibile definire due progetti con lo stesso nome.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un progetto.

Gestione: se nella metabase esiste il progetto con il nome specificato dall'utente, il sistema entra in edizione su quel progetto, altrimenti entra in edizione su un nuovo progetto.

C.2. CONTROLLI SUI SETTORI:

Descrizione: il nome di un settore non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un progetto.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome del settore il sistema segnala con un messaggio tale inconsistenza.

Descrizione: ogni settore appartiene ad un progetto.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: l'utente puo' definire un settore solo se ha gia' definito il progetto corrente.

Descrizione: i nomi dei settori appartenenti ad un progetto sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un settore.

Gestione: se nella metabase esiste il settore con il nome specificato dall'utente, il sistema entra in edizione su quel settore, altrimenti entra in edizione su un nuovo settore.

C.3 CONTROLLI SULLE CLASSI:

Descrizione: il nome di una classe non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare una classe.

Gestione: se l'utente non specifica, per errore e dimenticanza, il nome della classe il sistema segnala con un messaggio tale inconsistenza.

Descrizione: ogni classe appartiene ad un settore.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: l'utente puo' definire una classe solo se ha gia' definito il settore corrente.

Descrizione: i nomi delle classi appartenenti ad un settore sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare una classe.

Gestione: se nella metabase esiste la classe con il nome specificato dall'utente, il sistema entra in edizione su quella classe, altrimenti entra in edizione su una nuova classe.

Descrizione: le tre gerarchie di generalizzazione devono essere mutuamente esclusive.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente definisce una classe.

Gestione: se l'utente specifica, per la classe che sta definendo, due o più gerarchie simultaneamente il sistema segnala con un messaggio tale inconsistenza.

Descrizione: la classe antenata di una sottoclasse e' definita nello stesso settore della sottoclasse.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando si memorizza la definizione.

Gestione: se la classe antenata specificata dall'utente non e' stata ancora definita nel settore corrente, il sistema la crea automaticamente dietro conferma dell'utente.

Descrizione: il valore medio della cardinalita' di una classe e' compreso tra il valore minimo e quello massimo.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: durante l'edizione della classe.

Gestione: se l'utente specifica per la cardinalita' della classe un valore medio minore del valore minimo o maggiore del valore massimo, il sistema segnala con un messaggio tale inconsistenza.

C.4 CONTROLLI SUGLI ATRIBUTI:

Descrizione: il nome di un attributo non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un attributo.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome dell'attributo il sistema segnala con un messaggio tale incoerenza.

Descrizione: ogni attributo appartiene ad una classe.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: l'utente puo' definire un attributo solo se ha gia' definito la classe corrente.

Descrizione: i nomi degli attributi appartenenti ad una classe sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un attributo.

Gestione: se nella metabase esiste l'attributo con il nome specificato dall'utente, il sistema entra in edizione su quell'attributo, altrimenti entra in edizione su un nuovo attributo.

C.5. CONTROLLI SULLE ASSOCIAZIONI:

Descrizione: il nome di un'associazione non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'associazione.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome dell'associazione il sistema segnala con un messaggio tale inconsistenza.

Descrizione: una associazione ha specificato la classe dominio.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: L'utente può definire un'associazione solo se ha già definito la classe corrente e cioè il dominio.

Descrizione: una associazione ha specificato la classe codominio.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'associazione.

Gestione: se l'utente non specifica la classe codominio dell'associazione, il sistema segnala con un messaggio tale inconsistenza.

Descrizione: I nomi delle associazioni che pongono in relazione uno stesso dominio e codominio sono diversi tra loro.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'associazione.

Gestione: se nella metabase esiste l'associazione con il nome specificato dall'utente, avente come dominio la classe corrente e come codominio la classe specificata, il sistema entra in edizione su quell'associazione, altrimenti entra in edizione su una nuova associazione.

Descrizione: La classe codominio di una associazione e' definita nel settore corrente.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando si memorizza la definizione.

Gestione: se la classe codominio specificata dall'utente non e' stata ancora definita nel settore corrente, il sistema la crea automaticamente dietro conferma dell'utente.

Descrizione: la specifica della diretta e dell'inversa di una associazione e' consistente. Nella classe codominio e' definita un'associazione con il nome uguale a quello dell'inversa specificata nell'associazione corrente.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando si memorizza la definizione.

Gestione: quando l'utente definisce un'associazione, il sistema crea automaticamente, per la classe codominio, un'associazione avente il nome dell'inversa specificata e avente come codominio la classe corrente.

Descrizione: una associazione ha specificata l'inversa.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente definisce un'associazione.

Gestione: se l'utente non specifica l'inversa dell'associazione, il sistema segnala con un messaggio tale inconsistenza.

C.6. CONTROLLI SUI TIPI:

Descrizione: il nome di un tipo non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un tipo.

Gestione: se l'utente non specifica, per errore e dimenticanza, il nome del tipo il sistema segnala con un messaggio tale inconsistenza.

Descrizione: i nomi dei tipi sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un tipo.

Gestione: se nella metabase esiste il tipo con il nome specificato dall'utente, il sistema entra in edizione su quel tipo, altrimenti entra in edizione su un nuovo tipo.

C.7 CONTROLLI SULLE OPERAZIONI:

Descrizione: il nome di un'operazione non può essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'operazione.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome dell'operazione il sistema segnala con un messaggio tale incongruenza.

Descrizione: ogni operazione appartiene ad un settore.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: l'utente può definire un'operazione solo se ha già definito il settore corretto.

Descrizione: i nomi delle operazioni appartenenti ad un settore sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'operazione.

Gestione: se nella metabase esiste l'operazione con il nome specificato dall'utente, il sistema entra in edizione su quell'operazione, altrimenti entra in edizione su una nuova operazione.

C.9 CONTROLLI SULLE ATTIVITA':

Descrizione: il nome di un'attivita' non puo' essere la stringa vuota.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'attivita'.

Gestione: se l'utente non specifica, per errore o dimenticanza, il nome dell'attivita' il sistema segnala con un messaggio tale inconsistenza.

Descrizione: ogni attivita' appartiene ad un settore.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: implicita.

Gestione: l'utente puo' definire un'attivita' solo se ha
gia' definito il settore corrente.

Descrizione: i nomi delle attivita' appartenenti ad un
settore sono diversi.

Tipo: consistenza.

Attivazione: automatica.

Esecuzione: quando l'utente richiede di editare un'atti-
vita'.

Gestione: se nella metatase esiste l'attivita' con il
nome specificato dall'utente, il sistema entra in
edizione su quell'attivita', altrimenti entra in edi-
zione su una nuova attivita'.

BIBLIOGRAFIA

- [Alavi,84] Alavi, M. - An assessment of the prototyping approach to information system development, Communications of the ACM 27 (1984) 556-562.
- [Albano,84] Albano, A., Baiardi, F., Castelli, D., Orsini R. e Santarini, R. - Il trattamento della dinamica in Galileo, Rapporto Tecnico Collana DATAID, N. 22, (1984).
- [Albano,85a] Albano, A. e Orsini R. - Basi di Dati, Boringhieri (1985).
- [Albano,85b] Albano, A. e Orsini R. - A software Engineering Approach to Database design: the Galileo Project, in: Computer-Aided database design: the DATAID Project, Albano A., De Antioellis V. e Di Ieva A. (eds), North-Holland, Amsterdam (1985).
- [Albano,85c] Albano, A., Navezio, M. e Scurto, S. - Un sistema per la specifica dei requisiti di applicazioni che usano basi di dati, in: Atti del congresso A.I.C.A., Firenze 1985.

- [Bacci,83] Bacci,A. e Briganti,I. - Sistema Informativo per un centro di calcolo erogatore di servizi: il disegno della base di dati, Tesi di laurea in Scienze dell'Informazione, Università di Pisa (1983).
- [Baraglia,83] Baraglia,R. e Signore,O. - Il Sistema Informativo del Servizio di un centro di calcolo: la gestione degli utenti, Rapporto interno C83-19 (1983).
- [Batini,84] Batini,C. e Ferrara F.M. - Uno strumento per il progetto di basi di dati su "Personal Computer", in: Atti del congresso A.I.C.A., Roma 1994, 287-304.
- [Brodie,82] Brodie, G.L. e Silva,E. - Active and Passive Component Modelling: ACE/ECM, in: Information System Design Methodologies: A comparative review, Cile I.W., Sci H.G., Verrijn-Stuart A.A. (eds), North-Holland, 41-93 (1982).
- [Bubenko,80] Bubenko, A. - Information Modelling in the Context of System Development, IFIP Congress 80, 395-411 (1980).
- [Ceri,83] Ceri,S. - Methodology and tools for data base design, North-Holland, Amsterdam (1983).

- [Chen,76] Chen, P. - The Entity-Relationship Model: toward a unified view of data, Transaction on Database System 1 (1976).
- [Lenzerini,84] Lenzerini, M. e Santucci, G. - L'utilizzo di uno strumento automatico di documentazione nella progettazione concettuale di basi di dati, in: Atti del congresso A.I.C.A., Roma 1984, 257-286.
- [Martin,82] Martin, J. - Development without programmers (1982).
- [Navazio,85] Navazio, M.C. - Un sistema per la specifica dei requisiti di applicazioni che usano basi di dati: la comunicazione con l'utente, Tesi di laurea in Scienze dell'Informazione, Università di Pisa (1985).
- [Scurto,85] Scurto, S. - Un sistema per la specifica dei requisiti di applicazioni che usano basi di dati: gestione e analisi delle specifiche, Tesi di laurea in Scienze dell'Informazione, Università di Pisa (1985).
- [Surya,83] Surya, E. Yadov - Determining an Organization's Information Requirements: a State of the Art Survey, SIGDDP - ACM: Data Base, vol. 14, num. 3 pag 3-21 (1983).

[Wasserman,80] Wasserman, A.T. - Information System Design Methodology, J.Am.Soc.Inf.Sci.31,1 (1980).

[Teorey,82] Teorey, T.e Fry, J.P. - Design of Database Structures, Prentice-Hall Inc., Englewood Cliff, New Jersey (1982).

[Yao,78] Yao S.B., Navathe S.E., Weldon J.L. - An integrated Approach to Logical DataBase Design, N.Y.U., Symp. on DataBase Design (1978).

Questo documento si compone di v+187 pagine.

Stampato Lunedì, 14 Aprile 1986 - 2:41:33 p.m.