

progetto:

a p r i l

Automatic Payment: Reliable, Interactive and Lithe

di

.AlzG:Wolf.



progetto realizzato da

Luparello Alessandro

presentato

come tesina per l'esame di

Ingegneria del Software

(corso tenuto dal prof. **U. Lo Faso** nell'aa. 1998-1999)

settembre 1999

a p r i l

progetto del sistema informativo relativo alla gestione di una macchina di pubblico utilizzo per il pagamento automatico di fatture di utenza.

Al36:Wolf

Luparello Alessandro
(Palermo, 11 Febbraio 1976)

matricola n. 0319099
U n i v e r s i t à d e g l i S t u d i d i P a l e r m o
Ingegneria Informatica

v.le Regione Siciliana S.E. n.726 90129 Palermo
email: alswolf@tiscalinet.it tel: 0347-2533285

MEKO

Questa tesina viene fornita
come buon esempio di uso del
Turbo CASE.

Per il Turbo CASE stesso e per le
altre parti della tesina va sempre
tenuto presente che:

- 1) esse potrebbero contenere inesattezze
- 2) esse sono in ogni caso perfettibili

QUINDI: ABBIAATE SEMPRE SENSO
CRITICO!

Buon lavoro *Antonio Lo Faro*

Indice sintetico

Contents

I	Analisi del Sistema	<i>System Analysis</i>	
1.	Ambiente di Sviluppo		4
2.	Analisi dei Requisiti		5
3.	Studio di Fattibilità		6
4.	Analisi del Sistema		7
II	Pianificazione	<i>Project Planning</i>	
5.	Risorse assegnate al progetto.....		24
6.	Resource Sheet		30
7.	Task Usage		31
8.	Resource Usage		36
9.	Gantt Chart		40
10.	Pert Chart (livello più astratto)		43
11.	Pert Chart		44
III	Analisi del Software	<i>Software Analysis</i>	
12.	Funzionalità del progetto		46
13.	Data Flow Diagrams		47
14.	Requirement Dictionary.....		87
15.	State Transition Diagram		95
IV	Progettazione del Software	<i>Software Design</i>	
16.	Modulazione del progetto		96
17.	Modulazione del software		102
18.	Archivi utilizzati		124
	Riferimenti bibliografici	<i>Reference</i>	129
	Indice		130

Parte Prima

ANALISI DEL SISTEMA

System Analysis

Ambiente di sviluppo

Sistema Operativo ©**Microsoft** **Windows 95**

Editor grafico-testuale ©**Microsoft** **Word 97**

Il progetto che viene presentato è stato sviluppato, tra l'altro, con l'ausilio di alcuni strumenti software propriamente dedicati all'ingegnerizzazione del software, strumenti che si sono rivelati estremamente utili, se non addirittura indispensabili, specie nel mantenimento della consistenza sia per l'analisi che per la progettazione.

Strumento *Upper CASE* (Computer Aided Software Engineering):

©**StructSoft** **turbo CASE /Sys Ver2.02** Trial version (45 days)

Strumento di pianificazione della progettazione

©**Microsoft** **Project 98** SR-1 Evaluation copy

Per quanto riguarda le componenti non software dell'ambiente di sviluppo, è stato utilizzato un comune Personal Computer, una stampante per la produzione della presente documentazione ed una connessione alla rete Internet, necessaria per le ricerche sui dispositivi hardware da utilizzare nella realizzazione pratica del progetto presentato.

Personal Computer con mP Intel Pentium MMX 200MHz, HD di 2.1GB e RAM di 32MB

Stampante Laser

Connessione ad Internet presso il Centro Universitario di Calcolo¹

¹ Il cui utilizzo ci è stato consentito per la gentile cortesia del personale dello stesso, cui non possono non andare i nostri ringraziamenti.

Analisi dei requisiti

Obiettivi del sistema

Obiettivo principale è quello di rendere il nuovo sistema di pagamento delle bollette facilmente utilizzabile dal pubblico, tanto da sostituire (nel breve periodo) il sistema attuale nelle abitudini degli utenti.

Vincoli temporali ed economici

La macchina che si vuole costruire ha potenzialmente un grossissimo impatto sociale ed una evidente funzione di pubblico servizio. E' preminente quindi il raggiungimento degli obiettivi preposti, con standard qualitativi elevati, nel rispetto di vincoli di tempo e spese sostenibili.

tempo a disposizione per la realizzazione del progetto: 2 mesi
budget economico²: 55 mln di lire

Requisiti di produzione (imposti dal committente)

La macchina deve essere in grado di:

- 1) istruire il cliente sul suo stesso uso e guidarlo nelle operazioni
- 2) essere in grado di raccogliere i dati della fattura
- 3) accettare moneta e dare resto
- 4) fornire la ricevuta del pagamento effettuato
- 5) conservare i dati e calcolare statistiche (numero clienti totali persentatisi, estremi operazione, tempo medio per operazione, informazione sui casi peggiori), su base giornaliera, mensile, annuale.

La macchina deve inoltre risultare robusta ed affidabile nell'interazione con i clienti, soprattutto nei momenti di particolare affollamento.

² Se la commissione è pubblica, si suppone essere questo (quello economico) l'aspetto più fortemente vincolante (si attingerebbe infatti al denaro pubblico).

Studio di fattibilità

Introduzione

Il problema che ci si pone è quello di progettare un sistema informativo capace di sostituire l'attuale sistema di pagamento delle fatture di utenza domestica.

L'ambiente nel quale il sistema si dovrà inserire è quindi quello –particolarmente delicato- del contatto diretto con l'utenza, nell'ambito dello svolgimento di operazioni di pagamento.

Vincoli di progetto sono essenzialmente di tipo economico (principalmente) e temporale (solo secondariamente), si desidera comunque un'elevata qualità del risultato ottenuto (in termini di prestazioni e di "gradimento" dell'utenza).

Alternative possibili

Possibile alternativa è sicuramente il mantenimento dell'attuale sistema di pagamento, con la conseguente convalida di tutti gli aspetti negativi evidenziati nell'analisi dettagliata del sistema corrente, presentata nel seguito.

Ogni altra soluzione sembra configurarsi come una variante del sistema progettato.

Descrizione del sistema

Si vuole progettare una macchina per lo sportello di pagamento automatico (privo cioè di operatore) delle fatture di utenza (come ad esempio telefono, acqua, gas, elettricità,...).

La macchina deve essere in grado di:

- 1) istruire il cliente sul suo stesso uso e guidarlo nelle operazioni
- 2) essere in grado di raccogliere i dati della fattura
- 3) accettare moneta e dare resto
- 4) fornire la ricevuta del pagamento effettuato
- 5) conservare i dati e di calcolare talune statistiche

Analisi costi-benefici

giustificazione economica

Benefici

1. Benefici per l'attività di mantenimento delle informazioni
 - Possibilità di raccogliere ed archiviare automaticamente i dati
 - Mantenimento delle informazioni completo e sistematico
 - Maggiore portabilità delle informazioni
2. Benefici per l'attività di recupero delle informazioni
 - Recupero più rapido delle informazioni
3. Benefici per le capacità ristrutturanti del sistema
 - Possibilità di modifica simultanea di intere classi di informazioni
4. Benefici per il controllo dei processi e delle risorse
 - Riduzione della forza-lavoro necessaria per il controllo dei processi
 - Maggiore possibilità di mantenere un continuo controllo dei processi e delle risorse disponibili.

Costi

1. Costi preliminari
 - Costi di consulenza
 - Costo del capitale
2. Costi d'avviamento
 - Costo del sistema operativo
 - Costo della gestione dell'avviamento
3. Costi associati al progetto
 - Costo per l'acquisto del software applicativo
 - Costo del personale, delle spese generali,... per lo sviluppo interno.
 - Costo per la preparazione della documentazione
 - Costo per la gestione dello sviluppo
4. Costi di esercizio
 - Costo per la manutenzione del sistema (essenzialmente hardware)
 - Consumi (elettricità, carta, inchiostro,...)

Si osserva che i costi da sostenere saranno ammortizzati nel medio periodo dai benefici introdotti, specialmente quelli inerenti al risparmio dovuto al ricollocamento di personale umano. Si ritiene che i costi associati alla manutenzione (costi di esercizio) siano inferiori a quelli del sistema attuale.

Valutazione dei rischi tecnici

fattibilità tecnica

1) rischi di sviluppo

Si ritiene che l'ambiente nel quale si andrà ad operare non sia eccessivamente ostico e complesso, i vincoli e le necessità evidenziate in modo esplicito dal committente si pensano quindi complete ed esaustive.

Si suppone rivestire un ruolo fondamentale il modo di interfacciarsi con l'utente³.

2) disposizione delle risorse

In sede di progetto si ritengono sufficienti le risorse (economiche, temporali, di mezzi e di personale) messe a disposizione per lo sviluppo.

Si raccomanda caldamente al committente la professionalità e la capacità dei consulenti esterni messi a disposizione del gruppo di sviluppo, data la strategica importanza dei loro interventi.

3) tecnologia

La tecnologia (hardware e software) disponibile sia per l'ambiente di sviluppo che per la realizzazione effettiva del progetto, si ritiene più che sufficiente.

Implicazioni legali

fattibilità legale

Non intervengono particolari fattori legali che possono ostacolare o mettere in difficoltà lo sviluppo e la realizzazione del progetto.

L'ambiente nel quale la macchina si troverà ad operare è sicuramente particolare (anche dal punto di vista legale) ma la normativa che lo regola è conosciuta ed ormai "assestata".

Maggiore attenzione sarà impiegata invece per tutti quegli aspetti legati al trattamento dei dati personali, in relazione alle recenti leggi (nazionali ed internazionali, specie a livello europeo o nordamericano) sulla tutela della privacy.

³ Ci si riferisca al paragrafo sull'interfacciamento.

Analisi del sistema

Stato attuale del sistema pagamento

La situazione attuale del sistema di pagamento delle fatture di utenza è quello, noto ai più, dello sportello postale o bancario, con la presenza di un operatore⁴ che svolge il compito (di per sé meccanico, monotono e probabilmente demotivante) di richiedere il denaro necessario ed autenticare l'effettuazione del pagamento.

Un tale sistema presenta alcuni svantaggi, generalmente affatto trascurabili, sia per l'utente sia per l'operatore allo sportello:

L'utente deve

1. Recarsi all'ufficio postale o bancario, nei giorni e negli orari in cui questi risultano essere aperti al pubblico (in genere poche ore al giorno e solamente nei giorni non festivi).

Questa appare una condizione eccessivamente vincolante per tutte quelle persone che non hanno la possibilità, ad esempio per motivi legati alla loro posizione professionale, di assolvere al pagamento delle proprie fatture negli orari prestabiliti né di delegare alcuno a questo compito.

2. Attendere personalmente il proprio turno, spesso in condizioni disagiati (sovraffollamento dei locali, tempi di attesa molto lunghi, difficoltà oggettiva a trovare posti a sedere).

Le condizioni cui si fa riferimento sono (cosa sicuramente nota a chiunque si sia trovato, almeno una volta, nella situazione di dover eseguire un pagamento ad un ufficio postale o bancario) purtroppo lo status normale, consueto e consolidato. E' innegabile che un'attesa in piedi, per più di un'ora, in un ufficio affollato e magari con un inefficiente ricambio d'aria, non sia proprio quello che si definisce una situazione desiderabile (se la si vuole continuare a considerare comunque accettabile).

I due punti ora esplicitati sono fortemente connessi l'uno all'altro e sono legati anche all'aspetto relativo all'operatore allo sportello, che verrà trattato di seguito. Tutti i disagi sembrano comunque derivare dalla volontà di concentrare nello stesso luogo (l'ufficio postale o bancario) e nello stesso tempo (generalmente appena un quinto dell'intera giornata) operazioni anche differenti tra loro, che richiedono professionalità, tempi di esecuzione ed attenzioni spesso molto diverse.

Il personale allo sportello deve

1. assolvere a compiti estremamente monotoni e ripetitivi, poco gratificanti e, anche per questo, spesso motivo di tensioni e stress.

L'accumularsi di stress protratto a lungo nel tempo, non può che influire negativamente sulla personalità dell'impiegato, sulla sua 'motivazione' professionale e –di conseguenza- sia sul suo rendimento lavorativo che sul modo di relazionarsi con il pubblico (ledendo, in tal modo, sia gli interessi dell'ufficio ove si trova a lavorare, sia quelli dell'utente stesso).

⁴ In realtà esistono alcuni dispositivi abilitati all'effettuazione del pagamento automatico delle fatture di utenza, ma –allo stato attuale- essendo di proprietà di una particolare ditta fornitrice di servizi, accettano esclusivamente fatture emesse dalla ditta proprietaria (un esempio sono le macchine per il pagamento automatico delle fatture Telecom, realizzate dalla *sigma* (www.sigma.com)).

Descrizione di massima del nuovo sistema

Il sistema da progettarsi deve essenzialmente gestire in modo automatico il pagamento delle fatture di utenza domestica.

Il sistema deve :

1. istruire l'utente e guidarlo nelle operazioni
2. raccogliere i dati della fattura
3. accettare denaro e dare resto
4. fornire la ricevuta del pagamento effettuato
5. conservare i dati e di calcolare determinate statistiche

Analisi degli obiettivi

Il sistema progettato deve essere robusto nell'interazione con il pubblico, deve poter sopportare grossi carichi di lavoro continuando a gestire il servizio in modo efficiente.

Deve poter monitorare il suo stato di funzionamento interno in modo da garantire una corretta terminazione delle operazioni iniziate, anche al sorgere di problemi o malfunzionamenti.

E' necessario tener conto di quanto possa favorire un riutilizzo della macchina anche in ambiti diversi da quello di progetto (che dovrà quindi essere il più possibile flessibile); è necessario inoltre pensare all'eventuale riutilizzo di moduli software.

Il sistema sviluppato dovrà poter assecondare (con semplicità) a modifiche inerenti il paese (e quindi la lingua) di utilizzo, le banconote e le monete supportate (immissione di nuove, eliminazione di vecchie), le fatture che è possibile gestire.

Impatto del nuovo metodo

Indubbiamente sollevare il personale da incarichi che poco o nulla interessano la professionalità personale, permettendo loro di dedicarsi ad aspetti più interessanti e, se vogliamo, più consoni alle proprie capacità ed aspettative, comporterebbe una maggiore gratificazione (a vantaggio dell'impiegato), un maggior rendimento ed una maggiore produttività (a vantaggio dell'ente o dell'istituto).

L'utente dell'ufficio postale o bancario trarrebbe se non altro il beneficio di poter disporre di maggior tempo e di una maggiore attenzione (da parte dell'impiegato) per trattare operazioni che maggiormente richiedono l'intervento del personale umano (che non sono sicuramente le bollette da pagare!).

L'utente che deve pagare le proprie bollette, potrà farlo in modo automatico, semplice, veloce e senza alcuna restrizione di tempo:

a) automatico

La macchina che si è progettata non richiede, infatti, alcun intervento del personale umano per quanto riguarda l'effettuazione del compito principale cui è dedicata (pagamento delle fatture d'utenza).

In realtà è possibile l'interazione con operatori esterni per quanto riguarda la lettura dei dati statistici immagazzinati negli archivi della macchina. Risulta essere invece necessario un periodico intervento di personale dedicato alla manutenzione della macchina quando si impone la

soluzione di un problema (quale, ad esempio, l'immissione di un nuovo rullo di carta, la sostituzione di una cartuccia scarica di inchiostro,...).

b) semplice

Perché sia facilmente utilizzabile da tutti, la macchina deve guidare l'utente con estrema chiarezza ed immediatezza, senza risultare complessa o astrusa. Particolare cura si è quindi voluto dedicare alla creazione di un'interfaccia utente semplice, affidabile ed amichevole (*user friendly*). E' inoltre sembrato doveroso, corretto e fondamentale predisporre la macchina ad un facile uso da parte di persone affatto abituate a convivere con apparecchiature meccanico-informatiche, persone che non riescano a leggere bene il testo sul monitor (ad esempio anziani, persone con problemi alla vista, persone che non sappiano leggere,...) o persone portatrici di handicap fisici non gravissimi⁵.

c) veloce

Lo sportello di pagamento automatico delle bollette progettato, risulta sicuramente veloce ed immediato nel suo utilizzo (grazie alla meticolosa attenzione dedicata all'interfacciamento con l'utente, v. punto b); inoltre si suppone che generalmente –tranne in momenti di eccezionale affluenza- non vi sia un numero troppo elevato di persone in coda (in virtù della mancanza di restrizioni temporali, v. punto d., e grazie ad una capillare distribuzione sul territorio delle macchine stesse⁶).

d) senza restrizioni di tempo

La macchina viene messa a disposizione del pubblico praticamente in ogni ora del giorno ed ogni giorno dell'anno.

In realtà la macchina potrebbe essere temporaneamente fuori servizio ed attendere l'intervento del personale atto alla manutenzione, oppure potrebbe essere momentaneamente disabilitata quando vi è una connessione da parte dell'operatore che preleva le statistiche⁷.

⁵ Si veda in proposito il paragrafo seguente (interfacciamento con l'utente).

⁶ Requisito utile per ampliare, ad un vasto raggio, i vantaggi della macchina. Si ritiene consigliabile predisporre con queste macchine almeno buona parte degli uffici postali (supponendo per questi una diffusa presenza sul territorio).

⁷ Questi piccoli inconvenienti, che non sembrano comunque poter destare malcontenti o preoccupazioni di sorta (basterebbe un check-up completo ed un prelevamento delle statistiche effettuati con periodicità ed in momenti sfavorevoli all'afflusso di utenti), possono essere quasi completamente rimossi se si volesse realizzare una sorta di monitoraggio continuato di tutte le macchine (per tale monitoraggio le macchine progettate risultano già predisposte, v. paragrafo sugli sviluppi possibili).

Interfacciamento con l'utente

Con il termine 'interfacciamento' non s'intende esclusivamente quello delle schermate sul monitor ma l'insieme complessivo degli aspetti d'interazione della macchina con l'utente. Secondo questa, ampia, accezione si vuole un interfacciamento chiaro, accessibile, esaustivo ed amichevole.

- a) L'aspetto esterno della macchina non deve allontanare il potenziale utente, essa non deve apparire ostica ed ostile, inaccessibile o complicata. La prima cosa che si vuole trasmettere all'utente è un rassicurante senso di semplicità: la macchina deve sembrare (non solo essere) di semplice utilizzo, amichevole e di aiuto.

Si vuole che l'utente percepisca immediatamente l'affidabilità e la semplicità della macchina, che riesca la vederla subito come mezzo cui potersi servire con tranquillità (se non altro, per evitare di dover mettersi in coda all'ufficio competente).

Per questo si vuole progettare una macchina con:

- uno schermo chiaro e nitido, a colori (pur nel rispetto dei potenziali utenti con problemi di daltonismo) ed abbastanza grande;
- un rivestimento esterno il più possibile semplice, con spigoli arrotondati e colori cui si possa far corrispondere sensazioni di "fiducia" ed "amichevolezza" (ad esempio l'azzurro, il verde o l'arancione carico ed in genere colori non troppo spenti né eccessivamente vivaci);
- numero limitato di pulsanti, aperture, cassettoni, ecc... e, quelli esistenti, presentati in modo da essere immediatamente individuati (e riconosciuti per l'utilizzo che se ne dovrà fare), pur in un'armonica disposizione con il resto della macchina.

- b) La macchina deve poter essere utilizzata anche da utenti portatori di handicap fisici, senza che diventi necessario l'intervento di una persona accompagnatrice che aiuti l'utente stesso.

Si desidera una macchina che non solo non si configuri come meccanismo discriminante verso queste categorie di persone, ma –anzi– che possa diventare un'alternativa più che valida (addirittura preferibile) all'attuale sistema di pagamento.

Perché ciò possa verificarsi si vuole predisporre la macchina in modo da:

- presentare lo schermo disposto ad un'altezza non eccessiva (ad esempio a partire da 130 cm dal livello del suolo) e lievemente inclinato verso il basso (ad esempio 15 gradi) perché possa risultare perfettamente leggibile dall'altezza di una persona seduta e non comporti disturbo per la percezione degli altri utenti.
- presentare un lieve incavo nella parte bassa (non oltre i 70 cm dal suolo e con una profondità limitata ai 20-30 cm) in modo che l'utente in carrozzella possa comodamente avvicinarsi al video, ai pulsanti ed agli sportelli (immissione denaro, monete o banconote) senza avere necessità di assumere scomode posizioni: l'utente può avvicinarsi alla macchina senza che questa si comporti come un ostacolo per le sue gambe.
- presentare pulsanti e sportelli facilmente accessibili:
 - non oltre 140 cm di altezza dal suolo per l'introduzione di monete.
 - non oltre 110 cm di altezza dal suolo per l'immissione della bolletta o delle banconote (la cui introduzione comporta una difficoltà lievemente maggiore rispetto quella che si ha per le monete).
 - non oltre 100 cm dal suolo per lo sportellino di raccolta del resto (in modo che l'utente, anche quello in carrozzina, possa guardarvi dentro e prelevare il denaro con disinvoltura).
 - non oltre 130 cm per i pulsanti, e per il prelevamento della ricevuta.

- c) La macchina non deve effettuare discriminazioni verso le persone che potrebbero avere difficoltà visive (se non eccessive) o di comprensione dei messaggi video.

Si vuole, quindi, che la macchina presenti:

- pulsanti immediatamente visibili (dimensioni non troppo ridotte, si consiglia un diametro di almeno 2 cm, e colori che “spezzino” con il rivestimento esterno della macchina).
- scritte ed immagini (sul video) ben nitide, con alti contrasti e sufficientemente grandi (non bisogna ricorrere quindi a scritte “microscopiche”).
- sequenze di immagini di supporto al testo dei messaggi, che possano essere di ausilio ad un’immediata comprensione del testo scritto.

Si vuole che l’utente si appresti alla lettura del messaggio testuale avendo già un’idea, abbastanza completa e chiara, dell’informazione che deve ricevere. Idea recepita con la naturalezza e l’immediatezza che solo le immagini e le animazioni riescono a raggiungere, tali sequenze video devono quindi risultare gradevoli ed esplicative.

- un particolare pulsante che abiliti la macchina alla produzione di indicazioni sonore di supporto a quelle grafico-testuali presenti sullo schermo (in pratica la pressione di questo pulsante consente all’utente di essere guidato anche mediante messaggi vocali preregistrati, che diverrebbero la “lettura” dei corrispondenti messaggi testuali).

Si fa osservare che non si è voluto rendere automatica la funzione del supporto audio per garantire la –giustificata– necessità di privacy e riservatezza degli utenti in relazione al trattamento di dati ed informazioni comunque personali.

Dare all’utente la possibilità di usufruire di un supporto audio (non imposto), qualora si presentassero difficoltà nella percezione delle informazioni video, ci è sembrato andare nella direzione di un allargamento della base dei possibili utilizzatori della macchina (ci si riferisce esplicitamente a chi avesse problemi visivi, ad esempio gli anziani, o di comprensione dei messaggi scritti, ad esempio stranieri o analfabeti).

Si consiglia che i messaggi preregistrati siano chiari, con le parole ben scandite, ma non meccanici. La persona non abituata a rapportarsi con un sistema di automazione troverebbe “confortante” ascoltare una voce il più possibile naturale, calda, decisa e rassicurante⁸ (in tal modo risulterebbe meno violento il “distacco” dall’attuale metodo di pagamento delle bollette).

- d) La macchina deve essere utilizzabile con semplicità e disinvoltura da parte degli utenti più disparati. Ci si mette quindi nelle condizioni in cui il potenziale utente non abbia alcuna conoscenza, non solo del mondo dell’informatica, ma -in generale- di macchine con le quali poter interagire (ad esempio distributori di bibite, sistemi di emissione automatica di biglietti ferroviari, ecc...).

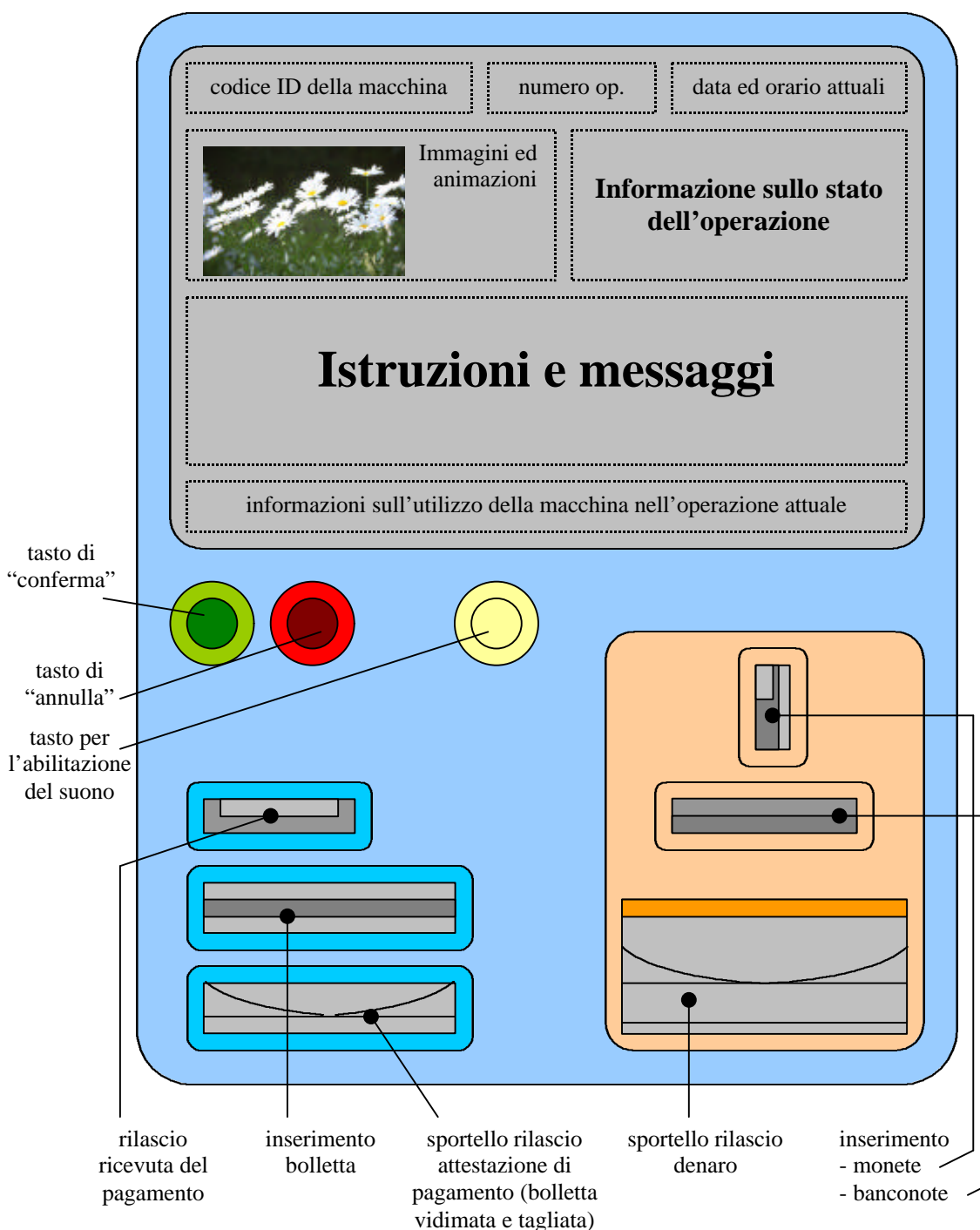
Per questa finalità sono necessari accorgimenti atti ad instaurare una comunicazione “amichevole” macchina-utente, si rivelano quindi senz’altro utili tutti gli aspetti già presi in considerazione precedentemente (design non freddo ed ostile della macchina, monitor nitido ed a colori, supporto con animazioni e suoni,...).

Si vuole fare osservare la necessità di adoperare, per i messaggi all’utente, un linguaggio chiaro, semplice ed immediato, senza fronzoli e privo di ambiguità.

⁸ Per questo scopo si consiglia l’introduzione di una voce femminile.

Tali messaggi devono servire a guidare l'utente in ogni fase della sua operazione, in modo da farlo sentire costantemente "guidato" (non deve fare uso di memoria, perché ha costantemente aiuti e suggerimenti sullo stato della sua operazione e su quello che deve fare).

Cosa di fondamentale importanza è non lasciare mai che l'utente possa trovarsi in difficoltà (situazione di "panico") ma supportarlo ogni istante ed in ogni situazione.



esempio pratico di interfaccia

Tecnologie necessarie

1. Per la fase di analisi non sono richieste particolari risorse tecnologiche (un semplice PC è sufficiente).
2. Le fasi di pianificazione e di progettazione del sistema necessitano invece di un ambiente di sviluppo efficiente, con strumenti (specialmente software) avanzati:
 - *Upper CASE* (Computer Aided Software Engineering)
 - Strumento di pianificazione della progettazione
3. Per la fase di codifica risultano ovviamente indispensabili computers (uno per programmatore, permettendo una codifica parallela di moduli indipendenti) e compilatori (nei linguaggi scelti).
4. La macchina necessita di taluni particolari **dispositivi hardware** per assolvere a determinate funzionalità esterne⁹:
 - dispositivo per il riconoscimento delle banconote

Sul mercato esistono diversi dispositivi con qualità anche molto simili tra loro. Caratteristiche discriminanti sono essenzialmente (in ordine di importanza):

- qualità del riconoscimento (numero parametri sui quali si basa)
- numero massimo banconote riconoscibili
- facilità di riprogrammazione delle banconote riconoscibili
- ingombro fisico e consumo elettrico

Si potrebbero suggerire –come esempio- due modelli italiani, entrambi della Coges s.r.l. (www.coges.it), ritenuti adatti alle necessità:

a) TP95

Caratteristiche Tecniche

Ingombro max: mm 100 x 120 x 170
 Peso: Kg. 1.5
 Alimentazione: 24 V. AC/DC
 Corrente max assorbita: 400 mA
 Potenza assorbita in lettura: 10 W
 Temperatura di funzionamento: 0°/+ 50°
 Numero max di banconote programmabili: 35
 Larghezza max banconote accettate: mm. 85
 Supporto lettore: Metallo
 Programmazione: Box program o PC

b) Eureka

Caratteristiche Tecniche

Ingombro max: mm 98 x 234 x 166.5
 Peso: Kg 3
 Alimentazione: 24 V. AC/DC
 Corrente max assorbita: 400 ma
 Potenza assorbita in lettura: 5 W
 Temperatura di funzionamento: 0°/ +40
 Numero max di banconote programmabili: 35
 Larghezza max banconote accettate: mm 73
 Supporto lettore: Alluminio
 Sistema di lettura: ottiche allo stato solido
 Accessori: Impacchettatore 600 banconote

- dispositivo per il riconoscimento delle monete

Anche in questo caso la scelta può basarsi su moltissime offerte del mercato. In genere tali dispositivi (riconoscitori di lega) effettuano il riconoscimento delle monete in transito attraverso una coppia di bobine captatrici che misurano parametri diversi, riferiti alla lega, allo

⁹ Di tutti i dispositivi hardware esplicitamente suggeriti si possono individuare le caratteristiche, anche più specifiche di quelle fornite in questo contesto, effettuando una semplice ricerca su Internet agli indirizzi relativi (che vengono sempre menzionati).

spessore e al diametro del conio, e l'immediato confronto con i valori inseriti in fase di programmazione.

A titolo semplicemente esplicativo (senza voler quindi imporre determinati modelli) si potrebbero suggerire due meccanismi prodotti dalla Coges s.r.l. (www.coges.it):

EUR/10

Collegabili in parallelo a tutti i distributori e al sistema E.C.S., accettano 11 monete (riprogrammabili) per 6 valori complessivi in uscita.

EUR/60

La risposta più evoluta per gestire la varietà dei coni in EURO degli 11 paesi dell'UME, compatibili con tutte le apparecchiature presenti sul mercato, accettano 60 diversi tipi di monete e gestiscono un numero crescente di valori a seconda del tipo di uscita prescelta.

Una particolare famiglia di modelli, quella contrassegnata dalla sigla MTB, sinonimo di tecnologia avanzata, include un nuovo protocollo di comunicazione internazionale.

Per l' EUR/60 MTB si hanno, ad esempio, le seguenti caratteristiche tecniche:

EUR/60 MTB

Caratteristiche tecniche:

ingombro max: mm 136, 64, 185
peso: g 380
alimentazione: V 24 AC/DC
consumo: W 3,6
numero monete accettate: 60
diametro monete accettate: mm 16-32
spessore monete accettate: mm 1-3.2
velocità accettazione: 3/sec
connessione: Seriale

- dispositivo di fornitura del resto

Sono dispositivi che si interfacciano sia con il deposito delle monete che con la cassa banconote. La loro funzione è quella di permettere il rilascio di denaro (resto dell'operazione di pagamento) all'utente.

Anche in questo caso, si pensa di suggerire (semplicemente come esempio) ancora un meccanismo prodotto dalla ditta italiana Coges s.r.l. (www.coges.it):

EURO 2000 MDB

Un dispositivo rendiresto tradizionale perfettamente compatibile con lo standard di comunicazione ICP/MDB che accetta 60 tipi di monete (si interfaccia con un EUR/60 MTB) ed eroga il resto attraverso 4 tubi di espulsione autonomi. Perfettamente modulare, può essere programmata tramite tastierino esterno o Box Program.

- dispositivo di acquisizione dell'immagine delle fatture

Si tratta di un comune scanner piano, posto lungo il meccanismo di spostamento della bolletta inserita. Tale lettore ottico deve avere una buona risoluzione in modo che dall'immagine acquisita possa essere estratta l'informazione necessaria (essenzialmente legata al codice a barre).

- dispositivo per la decodifica dei codici a barre

Questo è un dispositivo capace di decodificare, sulla base di dati contenuti in un archivio cui interfacciarsi, le informazioni codificate in barcode. L'archivio conterrà tutti gli strumenti atti a riconoscere ed a trattare il codice a barre letto, secondo le convenzioni di codifica (delle quali deve quindi essere a conoscenza).

Dopo una semplice ricerca, ci è stato possibile proporre (a titolo esemplificativo, non vincolante) un dispositivo in particolare che presenta le caratteristiche richieste:

ND50

L'ND50 è un decodificatore ad alta velocità, estremamente compatto, per lettori di codice a barre ad elevate prestazioni. Esso è collegabile tramite una linea seriale RS232C a qualsiasi tipo di host computer. E' inoltre possibile il collegamento in rete attraverso le interfacce seriali. E' completamente configurabile tramite una semplice tastiera oppure per mezzo ancora dell'interfaccia seriale.

Il Decodificatore ad alta velocità ND50 possiede 5 differenti modalità di lettura e 3 diversi modi operativi per soddisfare tutte le possibili esigenze di applicazione. E' disponibile in due versioni, con 8 Kbyte o 32 Kbyte di memoria con back-up. E' inoltre dotato di un real time clock per poter associare i dati temporali alle letture effettuate.

Specifiche Tecniche

Alimentazione/Consumo: tramite alimentatore esterno 12 Vol CC 1 Amp. / 300 mA.

Uscite: TTL compatibile o Open Collector.

Velocità: fino a 1.200 scan/sec.

Angolo: da 0° a 35° C.

Temperatura di Funzionamento/Immagazzinamento: da -20° a + 50° C / da -40° a + 70° C.

- dispositivo di taglio delle fatture

Trattandosi di uno strumento essenzialmente meccanico che consente di effettuare il taglio della fattura in modo da dividerla in due parti: quella che verrà immagazzinata dentro la macchina e quella che sarà restituita all'utente quale attestazione del pagamento.

- dispositivo di visualizzazione

Conformemente a quanto precedentemente descritto (interfacciamento con l'utente), tale monitor deve essere grande (almeno 15"), a colori e con una buona risoluzione grafica e testuale.

- dispositivo audio

Non sono necessarie, naturalmente, elevate potenze in uscita. Risulta invece molto utile una buona qualità del suono (che non deve essere metallico o meccanico).

Sviluppi possibili

A. Installazione di un gestore automatico di turno

E' possibile considerare la progettazione e l'introduzione di un dispositivo che consenta automaticamente la gestione del turno degli utenti.

Ciò verrebbe sicuramente in aiuto degli utenti stessi nei momenti di maggiore affollamento che, in questa sede, si ritengono comunque pochi e brevi¹⁰.

Possono esservi sicuramente diversi modi di assolvere a questo compito, suggeriamo alcune idee (cenni) per una gestione chiara ed "immediata", che non stravolga le abitudini degli utenti e non imponga loro particolari e complessi iter.

1. meccanismo di distribuzione di tickets numerati (di ormai diffuso utilizzo), display a led luminosi per la visualizzazione del turno attuale.

La macchina incrementa automaticamente il numero visualizzato, al termine di ogni operazione.

vantaggi:

- Gestione ordinata del turno nel caso di assenza di eccezioni.
- Possibilità, per l'utente, di allontanarsi dalla macchina (magari per svolgere altre commissioni) conservando il proprio turno.

svantaggi:

- L'automatizzazione implica una gestione rigida del turno, non consente alcuna flessibilità (come la necessità di anticipare il turno di una persona, magari disabile, senza sconvolgere l'intera gestione dell'avanzamento numerico o la gestione della situazione di "abbandono" del turno da parte di un utente).
- Necessità di controllare la disponibilità di tickets almeno giornalmente.

2. meccanismo dei tickets numerati e del display luminoso. E' l'utente –questa volta- che, arrivato alla macchina, incrementa manualmente (mediante un apposito pulsante) il display.

vantaggi:

- Gestione ordinata del turno.
- Possibilità, per l'utente, di allontanarsi dalla macchina (magari per svolgere altre commissioni) conservando il proprio turno.

svantaggi:

- Possibile confusione generata dal fatto che l'utente è diretto gestore del turno.
- Necessità di controllare la disponibilità di tickets almeno giornalmente.

3. meccanismo basato su tickets magnetici da inserire nella macchina. La macchina gestisce il turno automaticamente, attendendo l'arrivo del cliente selezionato per qualche secondo e chiamando poi il successivo. Non permette l'effettuazione di operazioni fuori dal proprio turno.

¹⁰ Concentrati, ad esempio, nell'ora immediatamente successiva la chiusura dei negozi o degli uffici. Una capillare distribuzione della macchina, specie nei quartieri più popolosi e più "attivi" economicamente, dovrebbe comunque ridurre moltissimo la possibilità di affollamento. Uno studio approfondito, magari con test o simulazioni, potrebbe evidenziare la necessità (o anche semplicemente l'utilità) di questa automatizzazione del turno oppure potrebbe stabilire che tale intervento non risulterebbe apportare vantaggi significativi all'utenza.

vantaggi:

- Gestione ordinata e sicura del turno, non c'è possibilità di "scavalcare" il controllo della macchina.
- Possibilità per l'utente, di allontanarsi con tranquillità dalla macchina (magari per svolgere altre commissioni) conservando il proprio turno.

svantaggi:

- Necessità di modifiche strutturali alla macchina per consentire la lettura ed il riconoscimento delle schede magnetiche.
- Gestione estremamente rigida del turno: impossibilità reale di effettuare operazioni al di fuori del proprio turno (non c'è la possibilità –quindi- che ad un disabile possa essere accordato diritto di precedenza¹¹).

B. Cambio valuta

Si vuole predisporre la macchina in modo che risulti estremamente facile ed immediato adattarla al funzionamento in mutate situazioni di valuta.

Oltre alla possibilità che essa venga utilizzata in paesi differenti (in questo caso basterebbe riprogrammare gli archivi delle banconote e delle monete) si prevede la possibilità che in uno stesso paese possa avvenire un cambio dell'unità monetaria (esempio evidente è il percorso che stanno seguendo diversi paesi europei per un'unificazione monetaria che porti gradualmente alla scomparsa delle valute nazionali ed all'introduzione dell'Euro). In questo caso deve essere possibile gestire la delicata fase di transizione (nella quale sono contemporaneamente utilizzabili entrambe le diverse valute) senza demandare all'utente alcuna operazione per "aiutare" la macchina.

Il progetto prevede infatti, proprio per introdurre il massimo livello di flessibilità, l'utilizzo –ad uso interno- di un valore numerico reale (floating point) cui far corrispondere il valore degli importi: una sorta di unità di misura indipendente da quella monetaria. La corrispondenza tra il valore delle monete/banconote inserite e la numerazione interna avviene mediante una semplicissima tabella di corrispondenza.

E' evidente che basta aggiungere opportunamente nuove righe a questa tabella per consentire l'attuazione delle funzionalità preposte. Un esempio di tale tabella potrebbe essere il seguente:

tipo banconota (dal riconoscitore)	valore economico associato	valore corrispondente (uso interno)
1	Lire 1000	0.5025
2	Lire 2000	1.0050
...
10	Euro 1	1.0000

¹¹ Limitazione che sarebbe possibile evitare prevedendo una tessera magnetica speciale (magari rilasciata gratuitamente dalle ASL a persone sofferenti o disabili) che venga riconosciuta dalla macchina. Questa abiliterebbe l'operazione di pagamento, bloccando l'avanzamento del turno, e restituirebbe la tessera all'utente.

E' necessario riprogrammare il riconoscitore di banconote o il riconoscitore di lega (modifiche effettuabili con le modalità relative ai dispositivi scelti; in genere molto semplici).

C. Cambio lingua

Si prevede che la macchina possa venire utilizzata anche in paesi diversi da quello dove esse viene progettata (Italia). I messaggi (testuali ed audio) agli utenti, dovendo essere di ausilio e non di intralcio ad una corretta effettuazione del pagamento, devono –naturalmente- essere espressi nella lingua del paese ospitante.

Questo avviene immediatamente, sostituendo semplicemente gli archivi nei quali vengono elencati i messaggi (testuali o audio; non quelli grafici, che non necessitano di “traduzione”).

D. Immissione di nuovi tipi di banconote o monete

Operazione necessaria quando la Zecca di Stato conia una nuova moneta o inventa nuove banconote. Anche questa è un'operazione semplicissima: basta introdurre nell'archivio delle monete o delle banconote (quello cui si è accennato precedentemente) il nuovo tipo (con il relativo valore economico ed il corrispondente valore numerico interno).

E' necessario riprogrammare il riconoscitore di banconote o il riconoscitore di lega (modifiche effettuabili con le modalità relative ai dispositivi scelti; in genere molto semplici).

E. Cancellazione di banconote o monete ritirate dalla circolazione

E' possibile che uno Stato ritiri dalla circolazione determinati modelli di banconote o di monete. Queste, divenute prive di valore economico, non devono poter essere utilizzate come strumento per il pagamento delle bollette. Basta effettuare una semplice modifica nell'archivio corrispondente (ancora quello cui si è fatto riferimento in precedenza) eliminando la riga relativa al tipo di banconota (o moneta) ritirata.

E' necessario riprogrammare il riconoscitore di banconote o il riconoscitore di lega (modifiche effettuabili con le modalità relative ai dispositivi scelti; in genere molto semplici).

F. Immissione di nuovi tipi di fatture di utenza gestibili dalla macchina

L'inserimento sul mercato di nuovi gestori pubblici o privati (di reti telefoniche o elettriche, ad esempio) potrebbe implicare la necessità di abilitare la macchina al trattamento anche dei nuovi tipi di fatture emesse.

Questo è fattibile con relativa semplicità. E' necessario modificare l'archivio delle maschere delle bollette (per permetterne il riconoscimento) e l'archivio di gestione (per la decodifica) dei codici a barre (ovviamente tale archivio può essere modificato aggiungendo nuovi records, sulla base di informazioni, necessarie al riconoscimento ed alla decodifica del barcode, fornite dal nuovo gestore).

G. Cancellazione dei tipi di fatture da non dover più gestire

E' ovviamente possibile effettuare l'operazione opposta a quella precedente. Necessita la modifica degli stessi archivi.

H. Connessione in rete locale

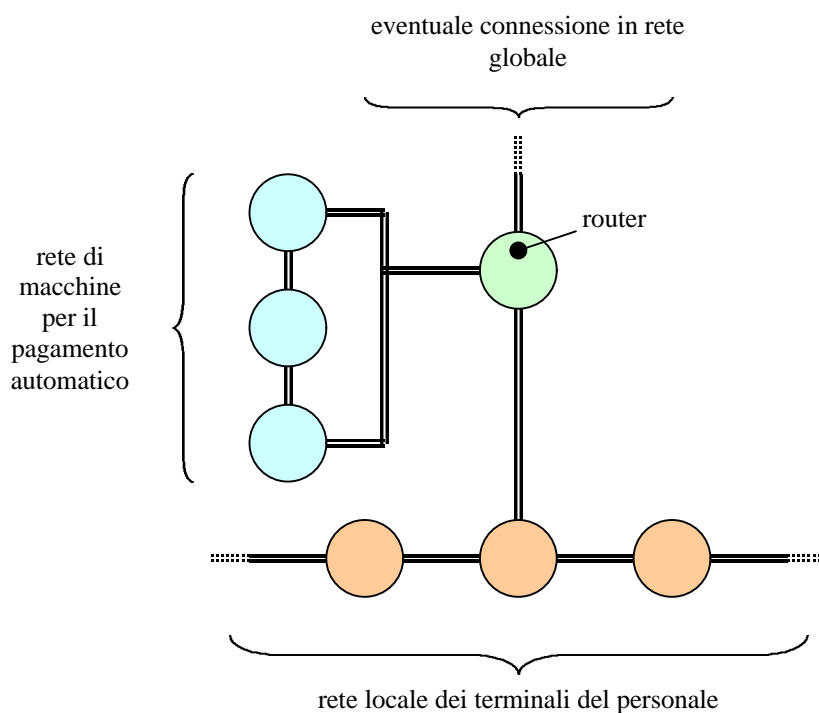
Si è voluto progettare la macchina in modo che risulti possibile una connessione in rete locale. Si immagina, in questa sede progettuale, una rete composta dalle varie macchine presenti in uno stesso locale (ad esempio un ufficio postale) e da terminali (utilizzati dal personale dell'ufficio).

Questa soluzione consentirebbe:

- L'estrazione contemporanea (ed automatizzabile, magari in relazione ad un temporizzatore) dei dati statistici raccolti ed immagazzinati.
- Un monitoraggio continuo delle macchine: una richiesta di intervento, avanzata in orari d'ufficio, potrebbe essere immediatamente soddisfatta.
- La riduzione al limite minimo del tempo di inutilizzazione della macchina per l'assolvimento alle pratiche di manutenzione (prelevamento statistiche, aggiornamento archivi).

Implicazioni realizzative:

- La connessione fisica in rete locale.



I. Connessione in rete globale

Si immagina di poter connettere direttamente in rete globale (ad esempio sfruttando la rete geografica Internet) le macchine stesse.

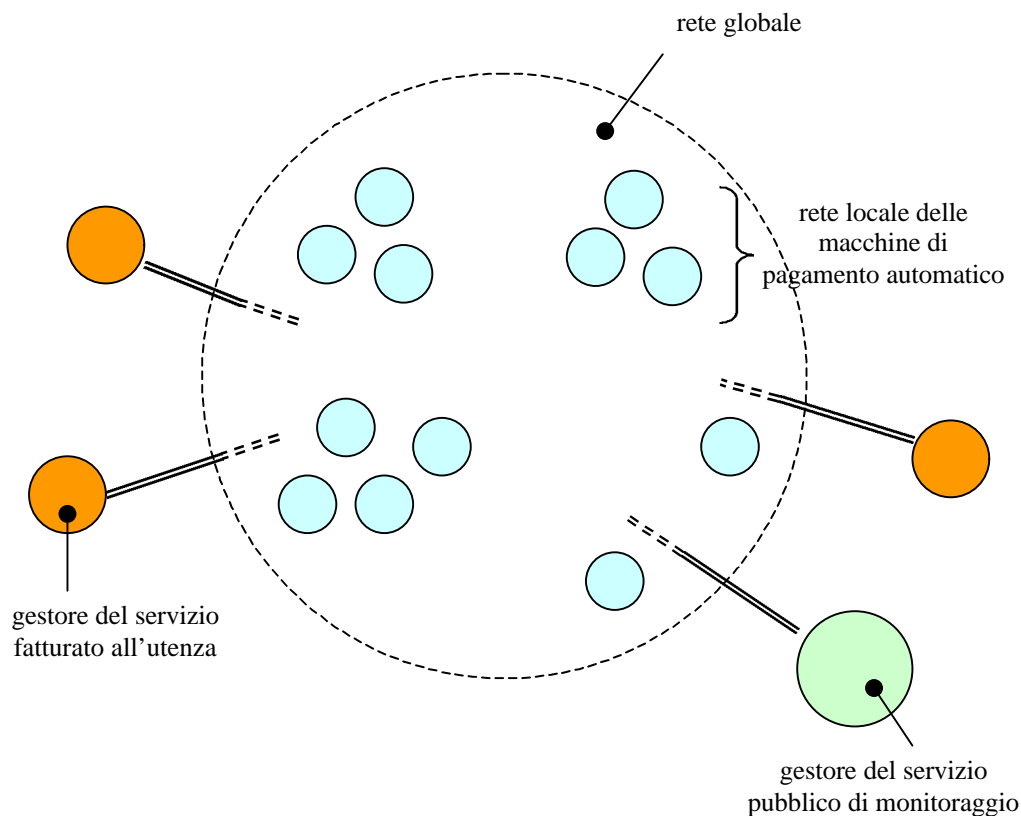
Questa soluzione consentirebbe:

- L'estrazione, via rete, delle informazioni statistiche memorizzate (senza la necessità di un operatore umano).

- Un automatico instradamento delle informazioni ai gestori dei servizi di cui si pagano le fatture (ad esempio, tutte le informazioni relative ai pagamenti delle fatture Telecom Italia potrebbero essere instadate, a richiesta o automaticamente a scadenze fissate, verso gli archivi centrali della Telecom stessa).
- Un monitoraggio continuato su tutte le macchine presenti in una città, in una provincia o a livello nazionale (o addirittura internazionale), con la possibilità di segnalazione immediata di eventuali richieste di manutenzione straordinaria.
- L'aggiornamento contemporaneo di determinati archivi presenti in tutte le macchine (ad esempio: l'introduzione di una nuova banconota non comporterebbe la modifica sistematica degli archivi-banconote manuale su tutte le macchine di una nazione !)

Implicazioni realizzative:

- Connessione fisica alla rete globale
- Strumenti software di gestione e controllo della consistenza dei dati negli archivi (le modifiche contemporanee devono avere realmente effetto contemporaneo).
- Strumenti software per la protezione dei dati (chiavi di accesso, privilegi, priorità,...).



Parte Seconda

PIANIFICAZIONE

Project Planning

Risorse assegnate al progetto

Risorse umane

Al progetto vengono assegnate diverse entità, con profili professionali e capacità differenti, con lo scopo di creare un team valido e produttivo, nel quale ogni figura abbia un proprio ruolo ed un proprio spazio e possa dare un contributo utile all'avanzamento del progetto stesso.

Le professionalità che si è voluto inserire si suppongono essere sufficienti alla realizzazione dell'intero progetto (nelle sue varie fasi); per quel che riguarda alcune figure specialistiche, il cui apporto è considerato indispensabile ad una buona analisi del problema¹, si è scelto di sfruttare la loro consulenza esterna (rimangono quindi al di fuori del team ristretto di lavoro).

Consideriamo brevemente le risorse umane facenti parte del team di lavoro:

1. Responsabile Progetto

E' una figura dai compiti preminentemente organizzativi e di supervisione. Viene nominato direttamente dall'organismo dirigente della casa di produzione (presso la quale il team, ed il progetto, sono inseriti), quale responsabile dell'intero progetto.

Funge quindi da intermediario privilegiato tra il team di sviluppo ed il committente, cercando di far collimare il più possibile le idee e le richieste di entrambi i lati.

Ad esso è assegnata la responsabilità della riuscita o del fallimento del progetto assegnato.

2. Analista

Si tratta di una delle più importanti professionalità all'interno del team. Il compito assegnato è quello di analizzare il problema da affrontare, cercando di studiarlo in ogni dettaglio (anche con l'aiuto di esperti nel settore e di consulenti esterni) carpandone gli aspetti fondamentali e sapendo trascurare gli aspetti ininfluenti ai fini del progetto. Il risultato della fase di analisi sarà la stesura di documentazioni corrette, complete ed esaustive sull'ambiente cui si andrà ad operare.

Per il team di sviluppo, l'analista diviene comunque il referente per tutte le interpellazioni sullo stato del problema, sull'efficacia e l'impatto delle soluzioni proposte, sull'affidabilità e l'opportunità di scelte ed assunzioni.

3. Ingegnere del Software

E' colui che ha la maggiore esperienza nella pianificazione e nella progettazione. Ad esso viene assegnato infatti il compito di pianificare l'intero progetto, nel rispetto dei vincoli imposti dal committente (sui quali vigila il responsabile del progetto), sia economici che temporali (soprattutto).

E' l'ingegnere del software che suddivide il problema in sottoproblemi, specificando i vari moduli software, le interfacce, le interconnessioni,... Ancora compito di questa figura è la ripartizione degli incarichi, la supervisione ed il coordinamento dei membri del team (che deve quindi almeno conoscere).

¹ Questo risulta evidente facendo riferimento proprio all'analisi del sistema.

4. Responsabile Codifica

Si è voluto inserire un coordinatore responsabile per la fase di codifica con l'esplicito scopo di rendere questa ancor più indipendente dalle altre fasi.

La presenza di una persona che si occupi di coordinare il gruppo di programmatori e di supervisionare al loro lavoro, che funga da referente per il responsabile del progetto sullo stato di avanzamento dei lavori di codifica, e da referente per i programmatori sull'attribuzione dei compiti, è sembrata un'idea vincente: sia per alleggerire la figura dell'ingegnere del software da compiti troppo specifici sia per dare una maggiore motivazione ai programmatori (è più che opportuno che tale responsabile abbia una conoscenza non superficiale dei programmatori stessi e che formi, con loro, un gruppo particolarmente affiatato; è inoltre opportuno che si instauri uno stretto rapporto anche con l'ingegnere del software).

5. Programmatori

Sono coloro che si occupano della realizzazione effettiva del codice eseguibile del software. Perché possa esservi un lavoro efficiente e soddisfacente è indispensabile l'affiatamento dei programmatori, sia tra loro che con il responsabile della codifica.

Nel rispetto dell'estro, delle attitudini e della professionalità del singolo individuo, ed ai fini di una produzione migliore, più celere e maggiormente efficiente ed efficace risulta molto utile che la suddivisione dei compiti ai vari programmatori tenga conto (nel modo maggiore possibile) proprio delle attitudini degli stessi (anzi la scelta stessa dei programmatori dovrebbe rispecchiare tale modo di ragionare).

Per far questo è necessaria quella conoscenza e quell'affiatamento, già sottolineati, con il responsabile della codifica.

Si vuole osservare che, perché si possa realizzare un lavoro qualitativamente molto soddisfacente, è sicuramente importante che all'interno del gruppo di sviluppo si realizzi presto (qualora non fosse già presente in precedenza; soluzione che –se possibile– sarebbe da preferire) una forte coesione e quell'affiatamento necessari ad un lavoro armonioso e piacevole (e, di conseguenza, più produttivo sia in termini di quantità che –soprattutto– di qualità).

Consideriamo adesso quelle figure professionali il cui apporto è ritenuto sicuramente importante ma che non risulta necessario inserire all'interno del gruppo di lavoro, ritenendo sufficiente semplicemente una loro consulenza esterna:

1. esperto di psicologia (comportamento e reazioni umane)

Tale figura professionale è stata considerata quale ausilio all'analisi ed alla progettazione dell'interfaccia utente della macchina che si intende realizzare. E' ritenuta utile per poter realizzare un prodotto che risulti usabile con estrema semplicità e disinvoltura dal pubblico (indipendentemente dallo status sociale, dal grado di istruzione o dalle capacità fisiche).

L'inserimento di consulenze con esperti nel settore denota la volontà di progettare meccanismi compatibili con l'ambiente (strutturale ed umano, soprattutto) circostante.

2. esperto di dispositivi hardware

La sua consulenza è ritenuta indispensabile quale ausilio sulla scelta della componentistica hardware che sarà presente nella macchina realizzata. Ricevute le indicazioni di massima da parte del team di sviluppo, sarà compito di questo esperto fornire una dettagliata analisi dei dispositivi più idonei, suggerendo anche strumenti valutativi per l'effettuazione della scelta più opportuna ed efficace.

3. esperto di affari legali

In ambiti di estrema delicatezza e dai confini ancora non molto marcati, come quello della creazione di software e del rapporto con il pubblico, un supporto legale è ritenuto non solo importante ma addirittura indispensabile.

Risorse Hardware

L'azienda nella quale il progetto viene ideato e sviluppato, mette a disposizione del team di sviluppo un laboratorio dove effettuare il lavoro. Il laboratorio si suppone confortevole, ben illuminato, ben arieggiato, dotato di servizi di buona qualità che possono rendere agevole e “comodo” il lavoro stesso.

Tale laboratorio sarà dotato di:

1. Zona riunioni

Si tratta di uno spazio, particolarmente confortevole, attrezzato per ricevere gli esperti nei diversi settori (consulenti esterni) ed il committente (o suoi rappresentanti fiduciari). In questo spazio si svolgono le riunioni del team, si prendono le decisioni per lo sviluppo, ci si incontra per i periodici momenti di revisione del lavoro.

2. Reparto computers

E' dove sono posti i computers per la fase di codifica (eventualmente a disposizione del gruppo solo in un secondo momento, quando dovrà iniziare proprio la fase di codifica). Deve essere presente però almeno un computer che contenga tutte le informazioni raccolte sul problema, la pianificazione del lavoro, lo stato attuale di avanzamento dei lavori, le interviste agli esperti.

In questo reparto è a disposizione, ovviamente, almeno una stampante di buona qualità (per la produzione della documentazione) e, possibilmente, scanner e masterizzatore.

3. Connessioni ad Internet

E' indispensabile un'efficiente connessione alla rete Internet: per l'effettuazione delle ricerche, per contattare (via E-Mail) i vari membri del team, i consulenti ed eventualmente il committente stesso, per poter relazionare a distanza con i consulenti o con il committente.

4. Fotocopiatrice e fax

Ritenuti validi supporti all'attività di stesura della documentazione ed all'interazione con l'esterno.

Saranno inoltre a disposizione del gruppo di lavoro:

5. Computers portatili

Si è scelto di assegnare computers portatili, invece di quelli tradizionali, a quelle figure che abbiano maggiore necessità di “indipendenza” dal laboratorio di sviluppo (ci si riferisce essenzialmente all'analista, che spesso potrebbe avere necessità di effettuare interviste o di osservare il problema da vicino, comunque all'esterno delle strutture dell'azienda).

6. Cellulari, registratori, materiale di consumo

E' ovvio che ogni membro del team di sviluppo deve essere contattabile in ogni momento (ed indipendentemente da dove si possa trovare), per questo motivo l'azienda mette a disposizione, per la durata del progetto, telefoni cellulari (eventualmente con vincoli sulle telefonate private).

Banalmente l'azienda provvederà a tutto il materiale di consumo necessario (CD, dischetti, carta,...).

Risorse Software

Sono a disposizione del team di sviluppo tutti i programmi e le applicazioni software ritenuti necessari o semplicemente utili. La scelta si basa su motivazioni aziendali strategiche (business partner, ambienti più promettenti,...) o, compatibilmente, su semplici suggerimenti da parte del team (ambienti più familiari,...).

Sono necessari²:

1. Un sistema operativo

Che risulti il più possibile stabile e sicuro. Deve supportare la distribuzione in rete.

Windows NT

©Microsoft Corporation

oppure un sistema operativo Macintosh (a seconda della filosofia aziendale).

Da tenere in considerazione il sistema operativo UNIX, forse più ostico degli altri già citati, ma sicuramente molto affidabile ed efficiente. Vincolo sull'utilizzo di UNIX è essenzialmente la presenza sul mercato delle applicazioni software richieste (v. seguito).

2. Un ambiente di navigazione in Internet

Internet Explorer 5.0

©Microsoft Corporation

o l'analogo Netscape (di utilizzo forse ancora più diffuso del citato explorer)

3. Un ambiente di video-scrittura

Che risulti semplice e potente nello stesso tempo, permettendo l'eventuale introduzione di grafici o l'importazione di documentazione o immagini dall'esterno.

Word 98

©Microsoft Corporation

Excell 98

©Microsoft Corporation

Acrobat 3.01

©Adobe Systems Inc.

²

Una motivazione ad alcune scelte ora espresse possono trovarsi nel paragrafo sull'analisi del software.

4. Un ambiente per la pianificazione

Che supporti la grande varietà di diagrammi, tabelle e schemi utili (spesso indispensabili) ad una corretta pianificazione. Tale strumento deve poter controllare la consistenza tra i vari diagrammi e deve permettere un flessibile e semplice aggiornamento dello stato attuale di avanzamento dei lavori.

Project 98

©Microsoft Corporation

5. Un ambiente di supporto all'analisi

Che supporti tutti quegli strumenti di documentazione grafica indispensabili alla comprensione del progetto stesso (come DFD, diagrammi architetture, dizionario dati,...). L'ambiente deve risultare molto efficiente, specie nella salvaguardia della consistenza.

TurboCASE ver 2.02

©StructSoft Inc.

6. Un ambiente di supporto alla progettazione

Quale può essere, ad esempio, un ambiente finalizzato alla costruzione immediata di prototipi di interfacce utente o un ambiente di sviluppo di specifiche TDN o simili.

7. Un ambiente per la codifica (compilatori)

Si basa sulla scelta che viene fatta in ambito progettuale sul linguaggio di programmazione più idoneo alla stesura del codice. Per motivi meglio descritti nell'analisi del software si sceglie il linguaggio di programmazione C++.

C++ Builder ver 3.0

©Borland

oppure il Visual C++, della Microsoft Corporation. Tale scelta viene lasciata al responsabile della progettazione.

project april

ID	Resource Name	Initials	Max. Units	Std. Rate	Ovt. Rate	Work	Cost
1	responsabile progetto	r	100%	L. 70.000/hr	L. 100.000/hr	115 hrs	L. 8.050.000
2	analista	A	100%	L. 50.000/hr	L. 80.000/hr	242 hrs	L. 12.100.000
3	programmatore	P	300%	L. 20.000/hr	L. 25.000/hr	286 hrs	L. 5.720.000
4	responsabile codifica	Rc	100%	L. 30.000/hr	L. 35.000/hr	96 hrs	L. 2.880.000
5	ingegnere del software	IS	100%	L. 65.000/hr	L. 100.000/hr	312 hrs	L. 20.280.000
6	esperto psicologia	Eps	100%	L. 250.000/day	L. 0/hr	26 hrs	L. 812.500
7	esperto affari legali	Eal	100%	L. 250.000/day	L. 0/hr	20 hrs	L. 625.000
8	esperto hardware	Eh	100%	L. 200.000/day	L. 0/hr	20 hrs	L. 500.000
9	committente	C	100%	L. 0/hr	L. 0/hr	52 hrs	L. 0

project april

ID	Task Name	Work	Duration	Details	30 Aug '99				
					S	M	T	W	T
1	analisi dei requisiti	84 hrs	3 days	Work		32h	28h	24h	
2	incontri preliminari con il committente	48 hrs	1,5 days	Work		32h	16h		
3	descrizione del prodotto	36 hrs	1,5 days	Work			12h	24h	
4	analisi e stesura delle specifiche	444 hrs	20 days	Work					40h
5	analisi preliminare del sistema	80 hrs	2 days	Work					40h
6	revisione con il gruppo di sviluppo	16 hrs	0,5 days	Work					
7	revisione iniziale con il committente	12 hrs	0,5 days	Work					
8	analisi raffinata del sistema	96 hrs	6 days	Work					
9	analisi della struttura hardware	40 hrs	2,5 days	Work					
10	specifica del sistema	144 hrs	9 days	Work					
11	specifiche hardware e software definite	0 hrs	0 days	Work					
12	revisione con il gruppo di sviluppo	32 hrs	1 day	Work					
13	revisione con il committente	24 hrs	1 day	Work					
14	specifiche del sistema approvate	0 hrs	0 days	Work					
15	progetto del sistema	109 hrs	16 days	Work					
16	pianificazione del progetto, stima costi-benefici	20 hrs	5 days	Work					
17	revisione del piano di progetto	10 hrs	1 day	Work					
18	progetto del sistema	63 hrs	8 days	Work					
19	revisione della struttura dei moduli	16 hrs	2 days	Work					
20	codifica e testing dei moduli	248 hrs	11 days	Work					
21	codifica dei moduli	92 hrs	4 days	Work					
22	testing dei moduli	156 hrs	7 days	Work					
23	progetto dell'interfaccia utente	98 hrs	16 days	Work					
24	pianificazione interfaccia	30 hrs	2,5 days	Work					
25	prototipazione interfaccia	32 hrs	11 days	Work					
26	tests amichevolezza	4 hrs	4 hrs	Work					
27	revisione con il committente	32 hrs	16 hrs	Work					
28	codifica dell'interfaccia	48 hrs	3 days	Work					
29	sviluppo interfaccia	48 hrs	3 days	Work					
30	approvazione interfaccia utente	0 hrs	0 days	Work					
31	integrazione, tests e validazione finali	138 hrs	7,5 days	Work					
32	integrazione moduli ed interfaccia	62 hrs	3,5 days	Work					
33	tests sul sistema	36 hrs	1,5 days	Work					
34	validazione finale	40 hrs	2,5 days	Work					
35	consegna finale del sistema	0 hrs	0 days	Work					

project april

[illegible]

project april

[illegible]

project april

[illegible]

project april

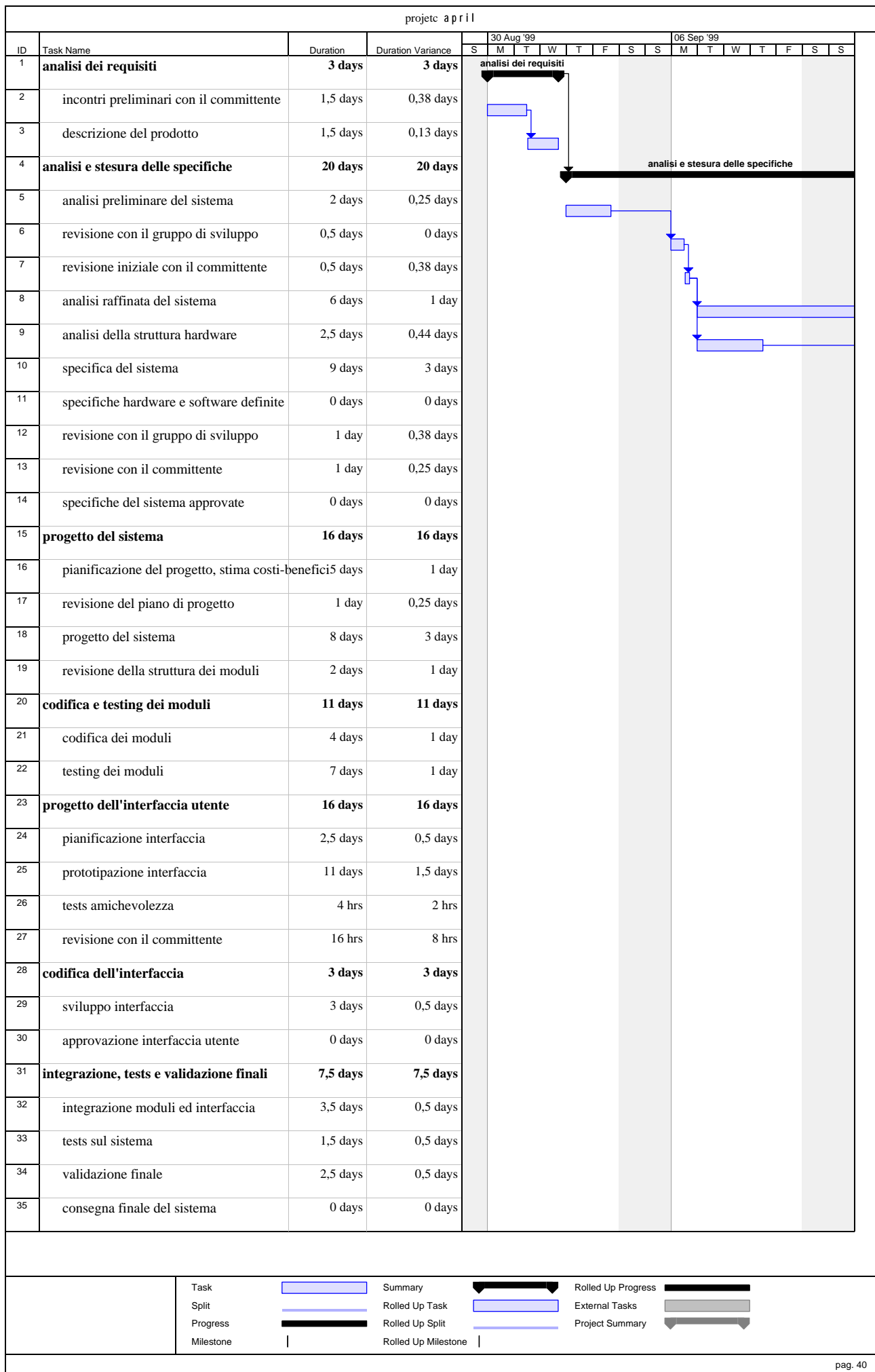
ID	Resource Name	Work	Details	30 Aug '99							06 Sep '99						
				S	M	T	W	T	F	S	S	M	T	W	T	F	S
	Unassigned	0 hrs	Work														
1	responsabile progetto	115 hrs	Work		8h	8h	8h	8h	8h			8h	8h	8h	4h		
	incontri preliminari con il committente	12 hrs	Work		8h	4h											
	descrizione del prodotto	12 hrs	Work			4h	8h										
	analisi preliminare del sistema	16 hrs	Work					8h	8h								
	revisione con il gruppo di sviluppo	4 hrs	Work									4h					
	revisione iniziale con il committente	4 hrs	Work									4h					
	analisi della struttura hardware	20 hrs	Work										8h	8h	4h		
	revisione con il gruppo di sviluppo	8 hrs	Work														
	revisione con il committente	8 hrs	Work														
	revisione del piano di progetto	3 hrs	Work														
	revisione con il committente	8 hrs	Work														
	validazione finale	20 hrs	Work														
2	analista	242 hrs	Work		8h	8h	8h	8h	8h			4h	8h	8h	8h	8h	
	incontri preliminari con il committente	12 hrs	Work		8h	4h											
	descrizione del prodotto	12 hrs	Work			4h	8h										
	analisi preliminare del sistema	16 hrs	Work					8h	8h								
	revisione con il gruppo di sviluppo	4 hrs	Work									4h					
	analisi raffinata del sistema	48 hrs	Work										8h	8h	8h	8h	
	specifiche del sistema	72 hrs	Work														
	revisione con il gruppo di sviluppo	8 hrs	Work														
	revisione con il committente	8 hrs	Work														
	revisione del piano di progetto	3 hrs	Work														
	progetto del sistema	15 hrs	Work														
	pianificazione interfaccia	10 hrs	Work														
	prototipazione interfaccia	10 hrs	Work														
	tests amichevolezza	4 hrs	Work														
	revisione con il committente	8 hrs	Work														
	tests sul sistema	12 hrs	Work														
3	programmatore	286 hrs	Work														
	codifica dei moduli	80 hrs	Work														
	testing dei moduli	140 hrs	Work														
	sviluppo interfaccia	36 hrs	Work														
	integrazione moduli ed interfaccia	30 hrs	Work														
4	responsabile codifica	96 hrs	Work														
	revisione con il gruppo di sviluppo	8 hrs	Work														
	revisione della struttura dei moduli	8 hrs	Work														
	codifica dei moduli	12 hrs	Work														
	testing dei moduli	16 hrs	Work														
	sviluppo interfaccia	12 hrs	Work														
	integrazione moduli ed interfaccia	28 hrs	Work														
	tests sul sistema	12 hrs	Work														
5	ingegnere del software	312 hrs	Work		8h	8h	8h	8h	8h			8h	8h	8h	8h	8h	
	incontri preliminari con il committente	12 hrs	Work		8h	4h											
	descrizione del prodotto	12 hrs	Work			4h	8h										
	analisi preliminare del sistema	16 hrs	Work					8h	8h								
	revisione con il gruppo di sviluppo	4 hrs	Work									4h					
	revisione iniziale con il committente	4 hrs	Work									4h					
	analisi raffinata del sistema	48 hrs	Work										8h	8h	8h	8h	
	specifiche del sistema	72 hrs	Work														
	revisione con il gruppo di sviluppo	8 hrs	Work														
	pianificazione del progetto, stima costi-benefici	20 hrs	Work														
	revisione del piano di progetto	4 hrs	Work														
	progetto del sistema	48 hrs	Work														
	revisione della struttura dei moduli	8 hrs	Work														
	pianificazione interfaccia	10 hrs	Work														
	prototipazione interfaccia	22 hrs	Work														
	revisione con il committente	8 hrs	Work														
	integrazione moduli ed interfaccia	4 hrs	Work														
	tests sul sistema	12 hrs	Work														
6	esperto psicologia	26 hrs	Work					8h	8h								
	analisi preliminare del sistema	16 hrs	Work					8h	8h								
	pianificazione interfaccia	10 hrs	Work														
7	esperto affari legali	20 hrs	Work					8h	8h			4h					
	analisi preliminare del sistema	16 hrs	Work					8h	8h								
	revisione con il gruppo di sviluppo	4 hrs	Work									4h					
8	esperto hardware	20 hrs	Work										8h	8h	4h		
	analisi della struttura hardware	20 hrs	Work										8h	8h	4h		
9	committente	52 hrs	Work		8h	4h						4h					
	incontri preliminari con il committente	12 hrs	Work		8h	4h											
	revisione iniziale con il committente	4 hrs	Work									4h					
	revisione con il committente	8 hrs	Work														
	revisione con il committente	8 hrs	Work														
	validazione finale	20 hrs	Work														

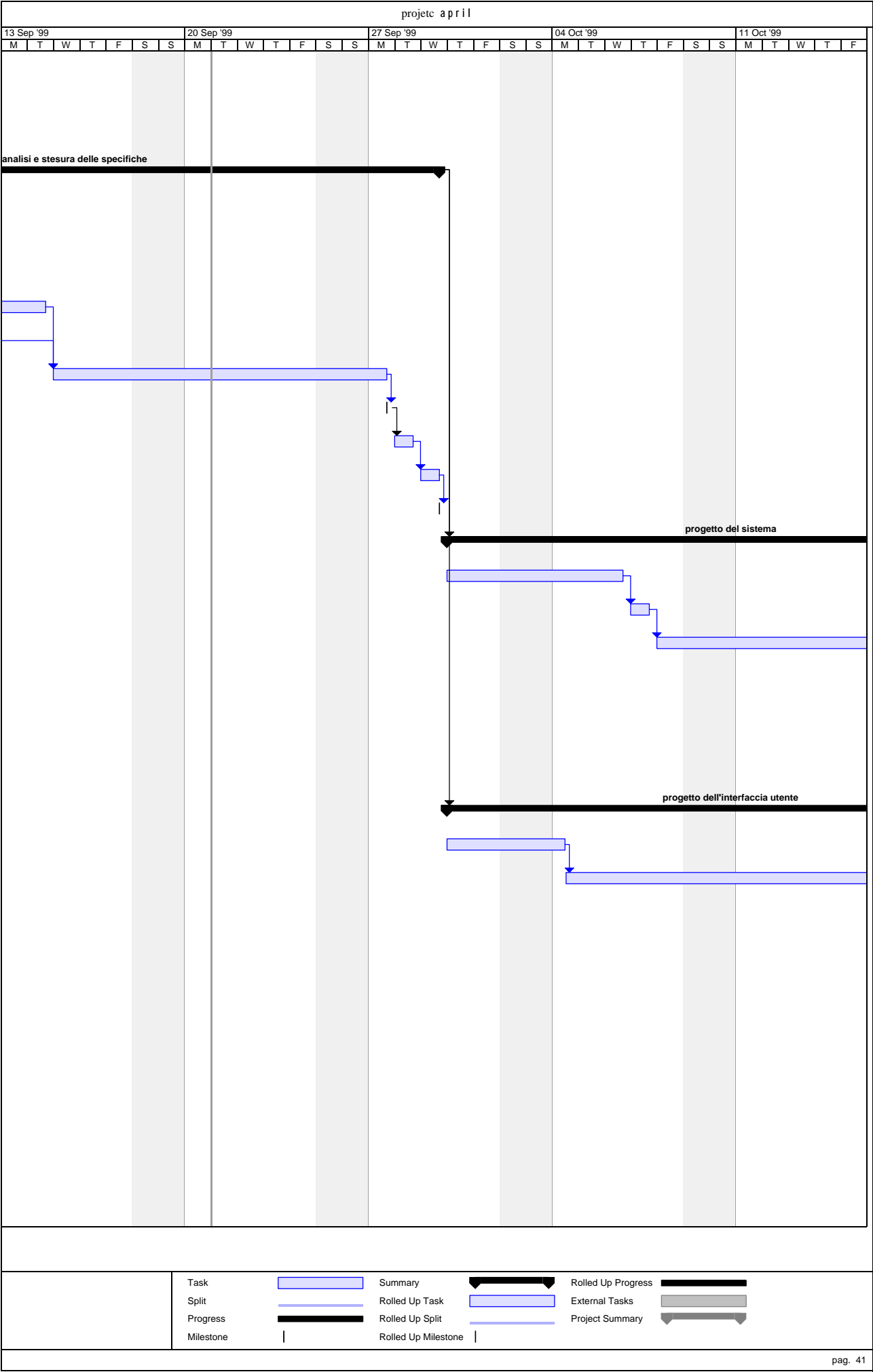
project april

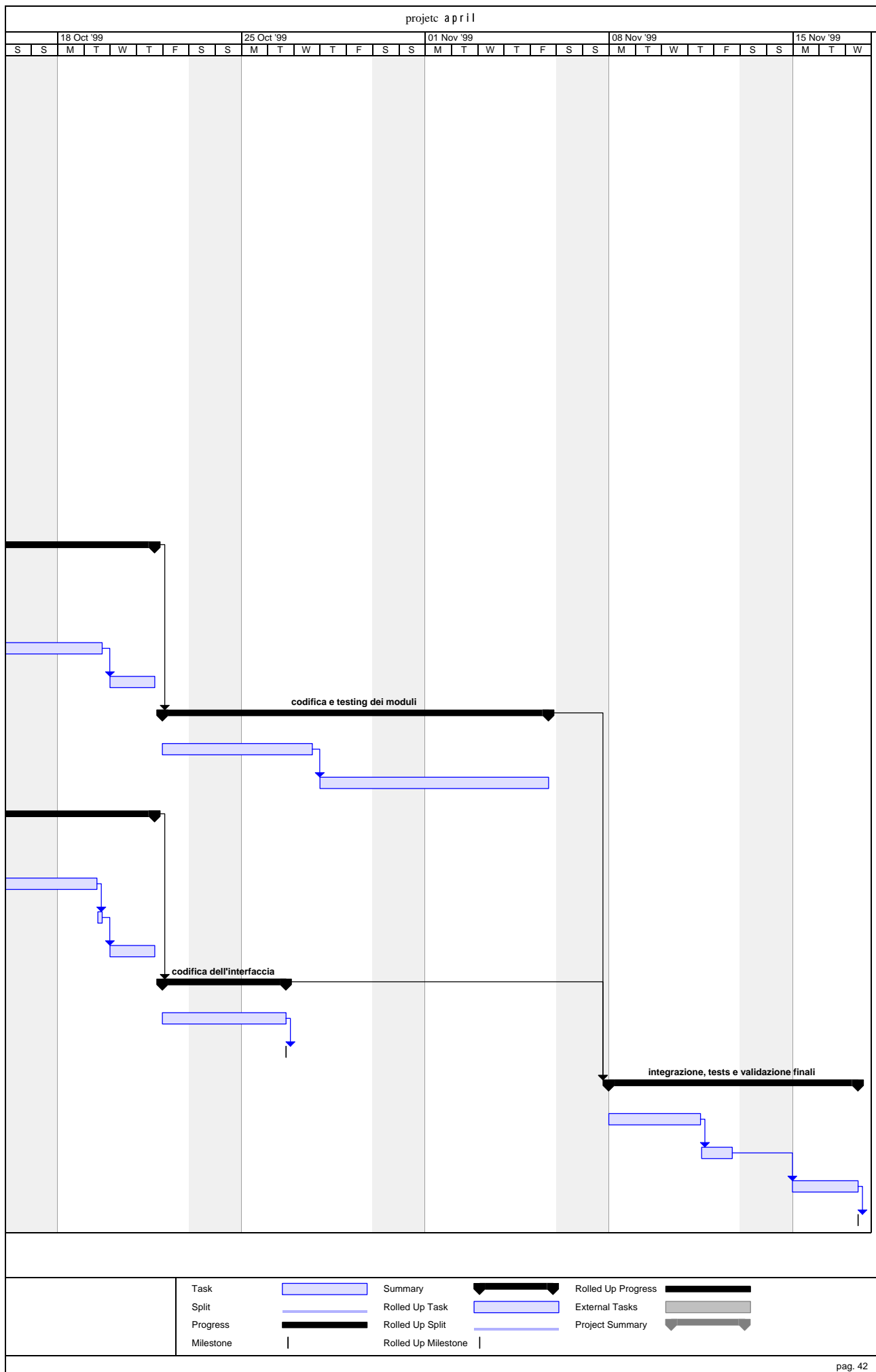
[illegible]

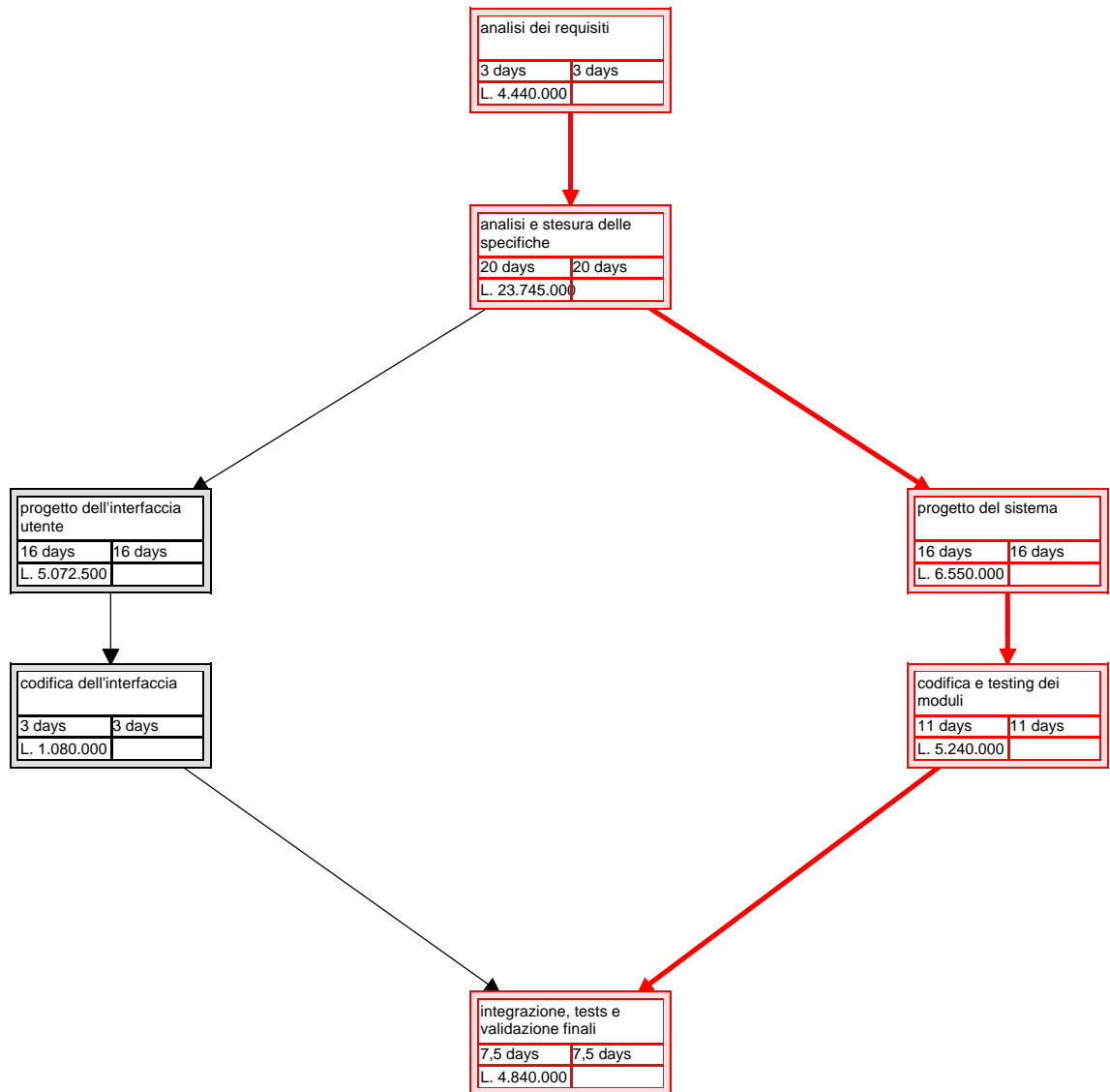
project april

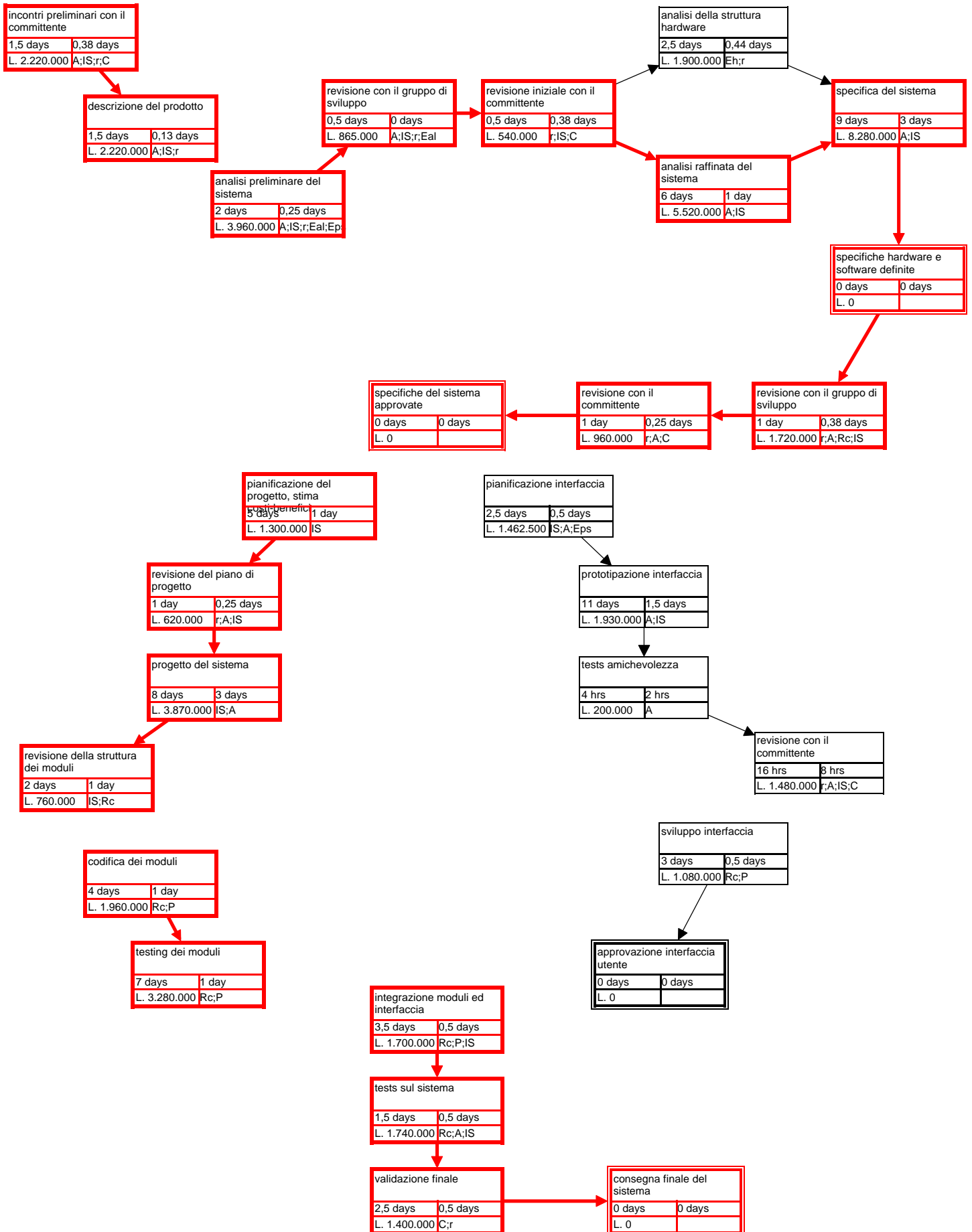
[illegible]











Parte Terza

ANALISI DEL SOFTWARE

Software Analysis

Funzionalità del progetto

La macchina che si intende progettare deve assolvere a diversi compiti, come descritto nell'analisi dei requisiti e nella pianificazione degli obiettivi. Al fine di ridurre la complessità nell'affrontare il problema, questo viene suddiviso in sottoproblemi logici (cui corrisponderà verosimilmente una suddivisione in moduli del software stesso):

A. Effettuazione dell'operazione di pagamento della bolletta

- Lettura dati della bolletta

Sottoproblema relativo al riconoscimento del tipo di bolletta inserita ed all'estrazione dei dati significativi (importo da pagare, identificativo del cliente,...).

- Operazione di pagamento

Sottoproblema che gestisce l'intera operazione di inserimento/emissione di denaro. Permette di inserire il denaro per il pagamento e gestisce "intelligentemente" la rilascio del resto.

- Trattamento della bolletta ed emissione della ricevuta

Sottoproblema che si interessa di gestire il trattamento della bolletta, la sua eventuale vidimazione e le coordinate del taglio, e l'emissione della ricevuta di pagamento (che potrebbe avere valore di credito per l'utente).

B. Gestione delle statistiche

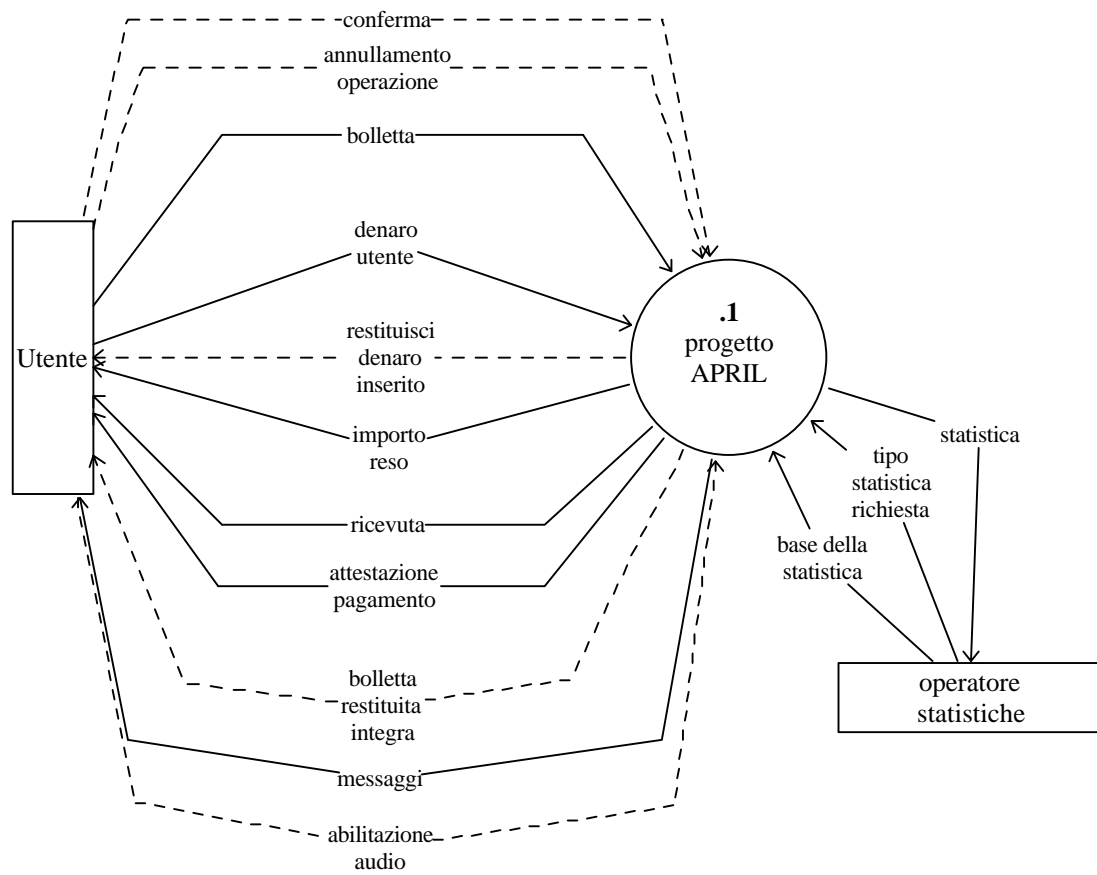
Sottoproblema relativo alla formattazione ed all'immagazzinamento dei dati raccolti ad ogni operazione. Si occupa di costruire le statistiche eventualmente richieste dall'operatore.

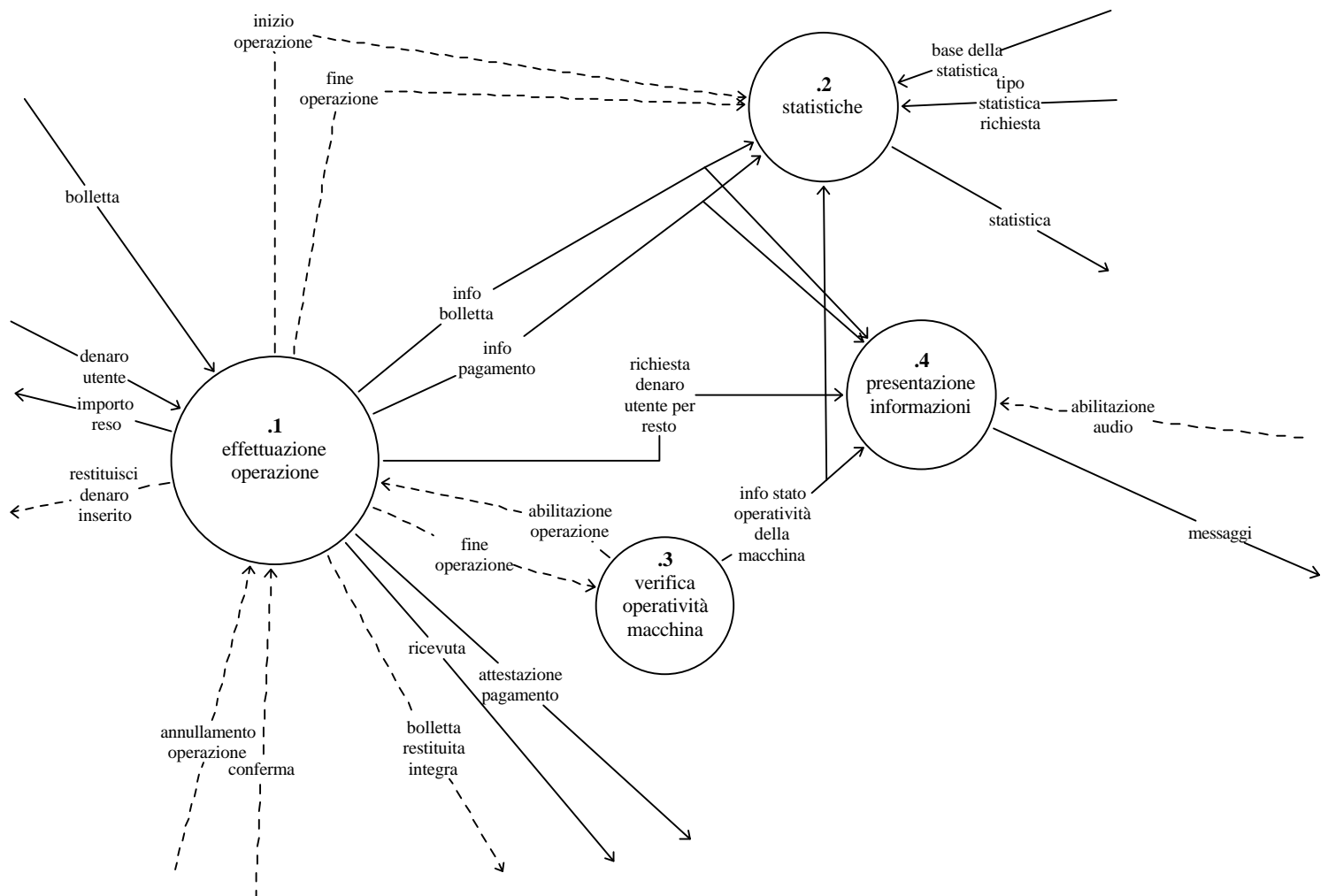
C. Interfacciamento con l'utente

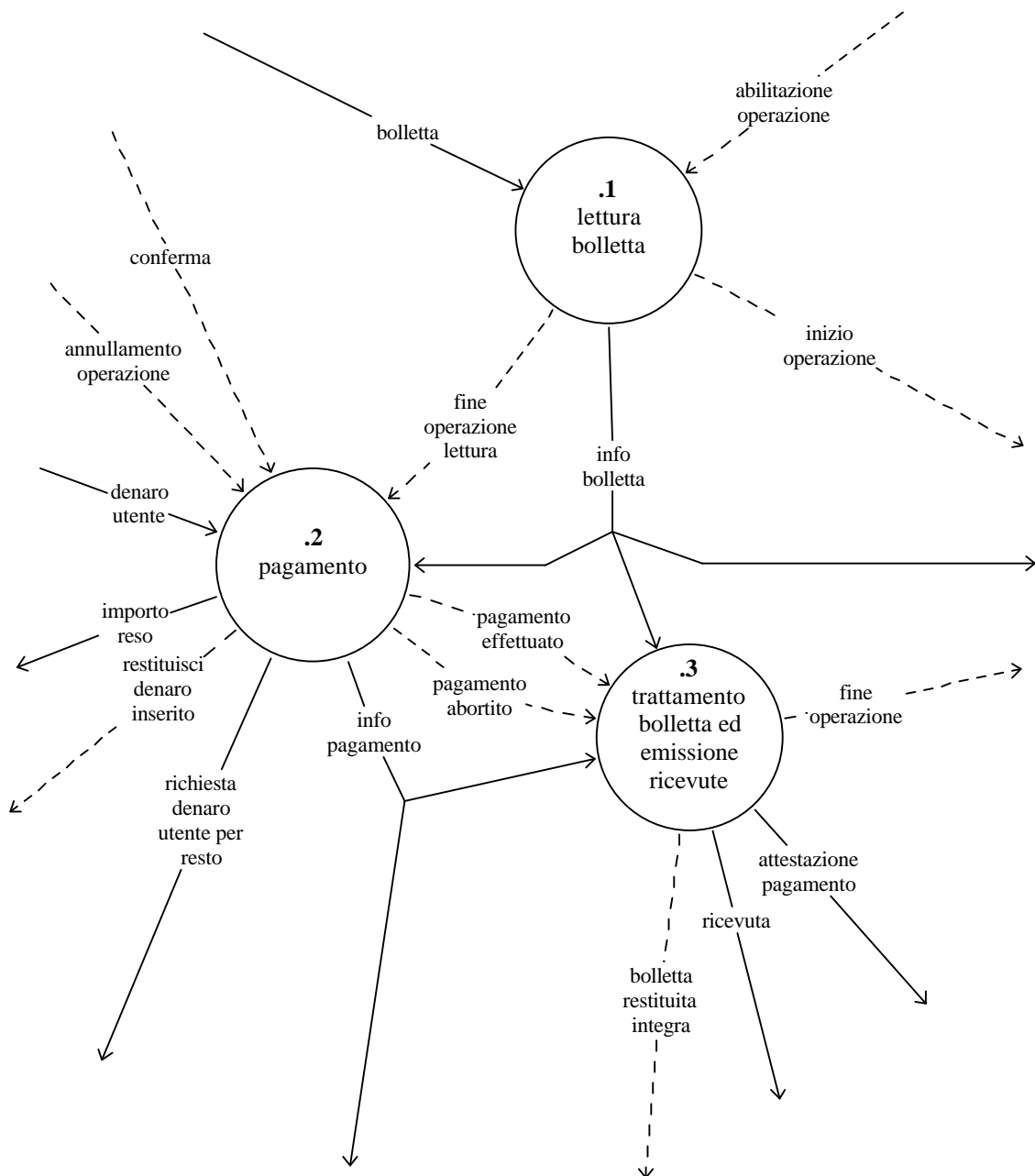
E' relativo al problema dell'interfacciamento esterno con il potenziale usufruttore della macchina. Si tratta di un sottoproblema molto delicato, sulla cui importanza si è molto insistito in sede di analisi (v. paragrafo dell'*interfacciamento con l'utente*).

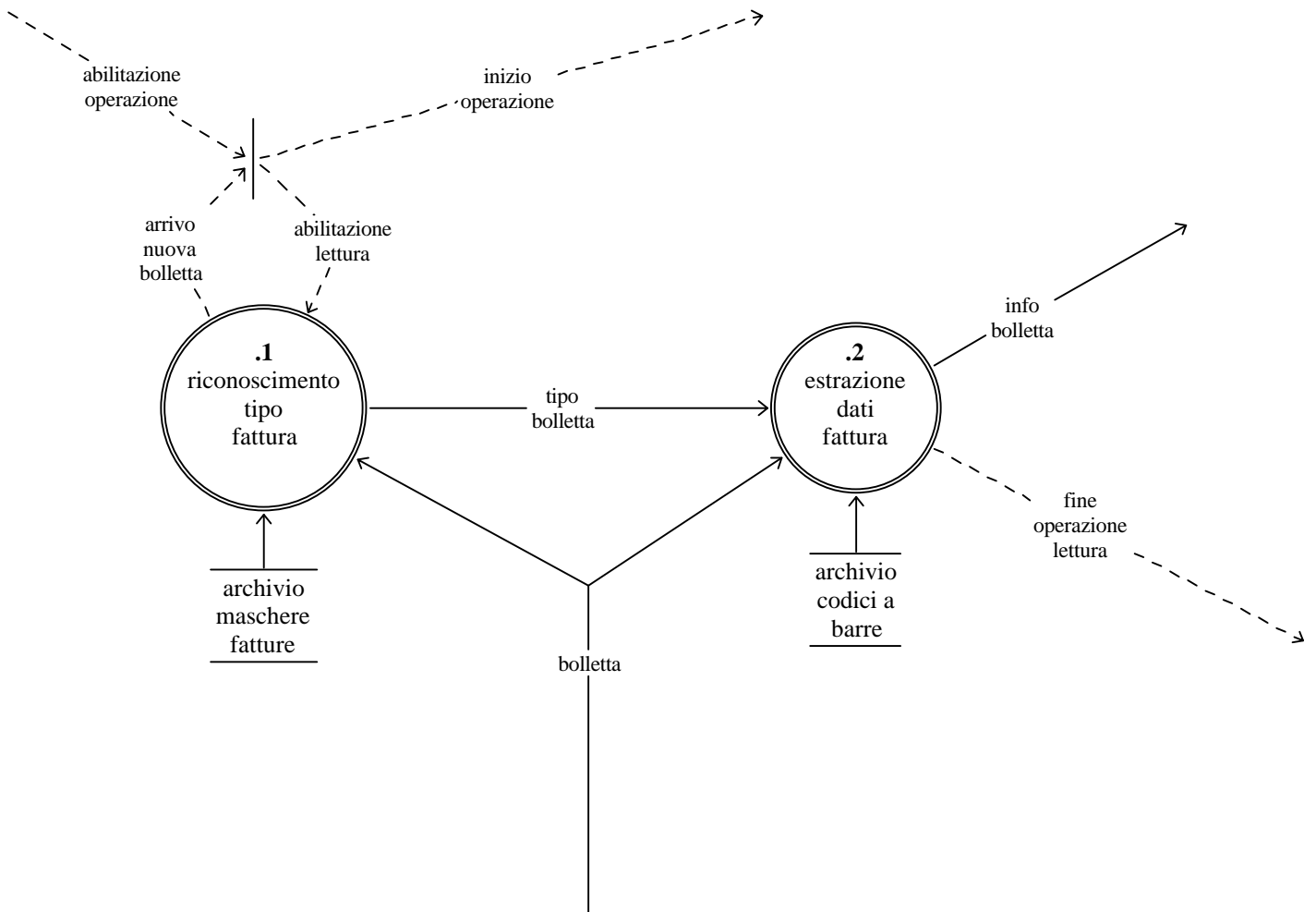
D. Verifica operatività della macchina

Sottoproblema che si occupa di garantire l'operatività della macchina per l'operazione immediatamente successiva. Il monitoraggio completo della macchina consente di bloccare l'utilizzo (al completamento dell'ultima operazione in corso) in uno stato sicuro, in attesa di intervento di manutenzione (che potrebbe essere richiesto esplicitamente dalla macchina stessa, considerando lo scenario di una sua connessione in rete, v. paragrafo su *sviluppi possibili*).









if RECEIVE:abilitazione_lettura = TRUE then

INPUT:bolletta

READ:archivio_maschere_fatture

PUT:tipo_bolletta

EMIT:arrivo_nuova_bolletta

Conviene lasciare lo scheletro
indicato in testa alla specifica.

Sarà poi possibile utilizzare nello
pseudo codice i dati definiti
nello scheletro.

if not (GET:tipo_bolletta = unknown)

then

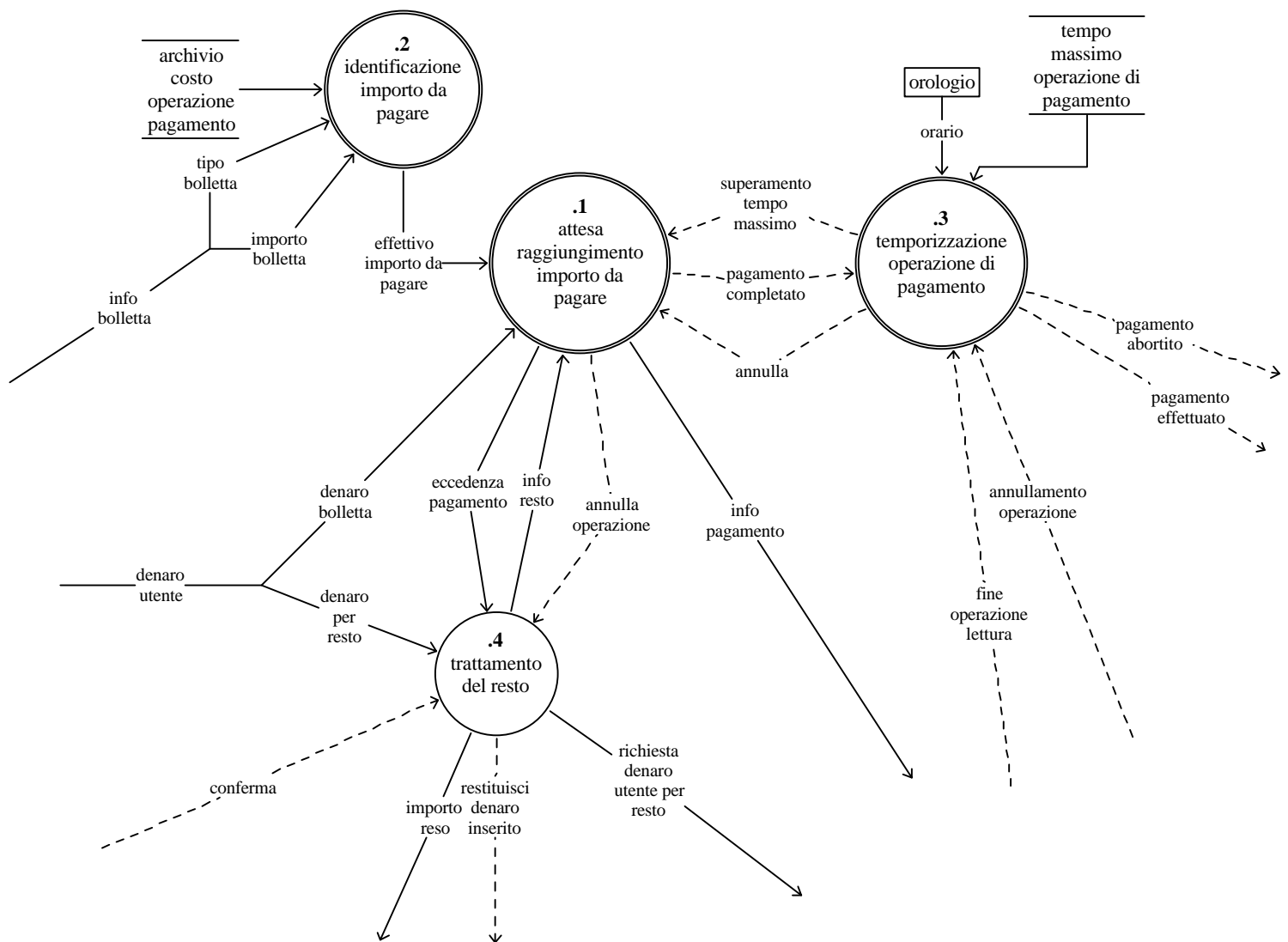
READ:archivio_codici_a_barre

INPUT:bolletta

OUTPUT:info_bolletta

EMIT:fine_operazione_lettura

Input		Process	Output	
arrivo nuova bolletta	abilitazione operazione	riconoscimento tipo fattura	inizio operazione	abilitazione lettura
TRUE	ABILITATA	1	TRUE	
OTHERS			FALSE	



if not (INPUT:tipo_bolletta = unknown)
then USE:tipo_bolletta

PRODUCE:effettivo_importo_da_pagare
INPUT:importo_bolletta
READ:archivio_costo_operazione_pagamento

if DETECT:annulla = TRUE or DETECT:superamento_tempo_massimo
then EMIT:annulla_operazione

PRODUCE:eccedenza_pagamento =
GET:effettivo_importo_da_pagare
INPUT:denaro_bolletta

USE:info_resto

usa le informazioni sullo stato del resto (ad es. immissione di altro denaro
per facilitare le operazioni di produzione del resto) aggiornando lo stato di
pagamento

Terminata l'operazione di pagamento si generano i dati in uscita

OUTPUT:info_pagamento

EMIT:pagamento_completato

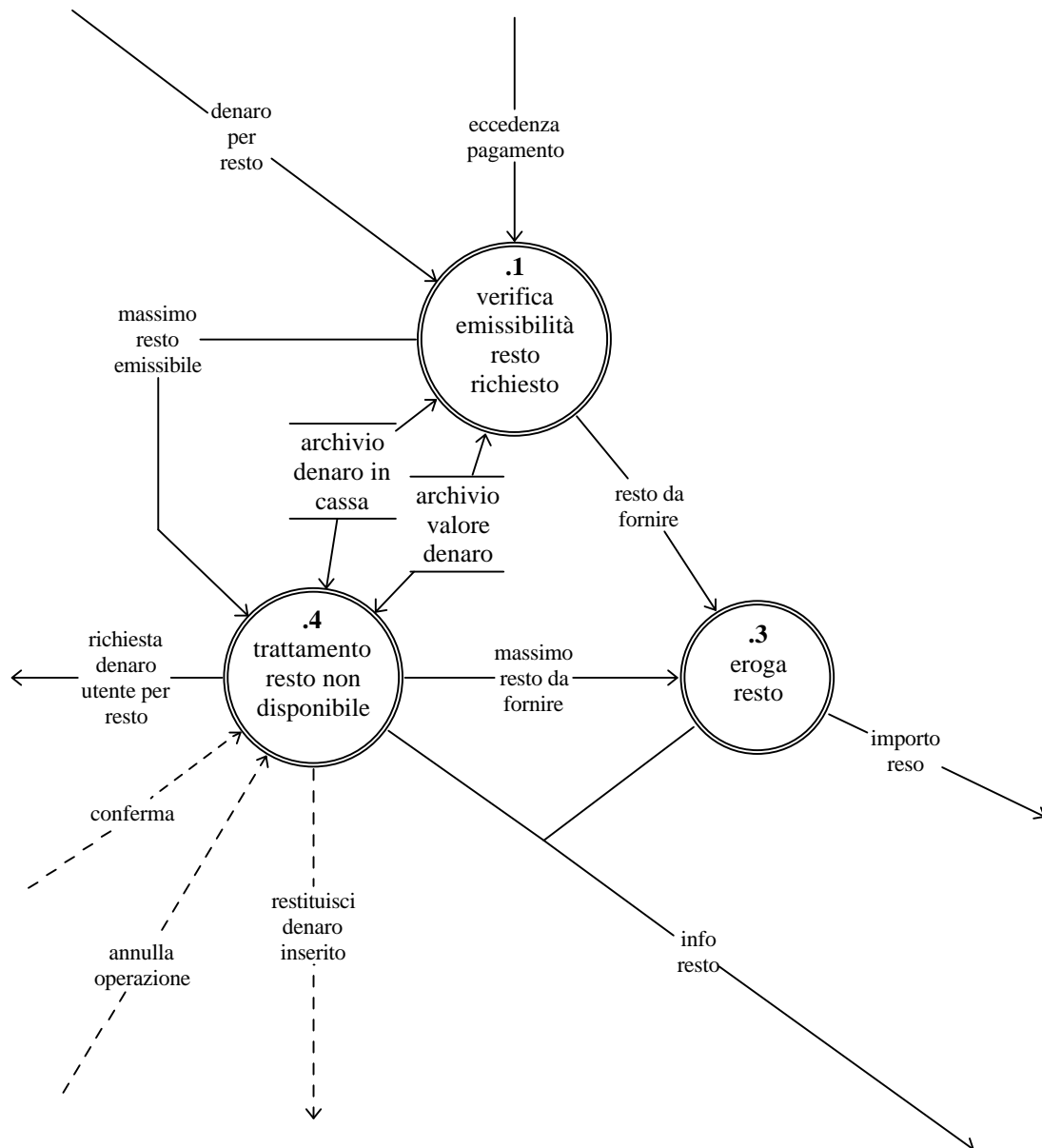
```
if DETECT: fine_operazione_lettura = TRUE  
then start = assoluto ( INPUT:orario )
```

```
if assoluto ( INPUT:orario ) - start >= READ:tempo_massimo_operazione_di_pagamento
```

```
then  
  EMIT: pagamento_abortito = TRUE  
  EMIT: superamento_tempo_massimo = TRUE
```

```
if DETECT: annullamento_operazione = TRUE  
then  
  EMIT: pagamento_abortito = TRUE  
  EMIT: annulla = TRUE
```

```
if DETECT: pagamento_completato = TRUE  
then  
  EMIT: pagamento_effettuato
```



READ:archivio_denaro_in_cassa

READ:archivio_valore_denaro

INPUT:eccedenza_pagamento

INPUT:denaro_per_resto

In base ai dati sulla quantità di denaro presente all'interno della macchina ed all'importo inserito dall'utente, si produce l'importo del resto da fornire (che potrebbe essere inferiore a quello dovuto)

if 'resto interamente fornibile' then PRODUCE:resto_da_fornire

else PRODUCE:massimo_resto_emissibi

```
if GET:resto_da_fornire
then
  OUTPUT:importo_reso = GET:resto_da_fornire

if GET:massimo_resto_da_fornire
then
  OUTPUT:importo_reso = GET:massimo_resto_da_fornire

OUTPUT:info_resto
```

READ:archivio_denaro_in_cassa

READ:archivio_valore_denaro

In base alla quantità di denaro presente nella macchina suggerisce all'utente l'inserimento di una certa somma di denaro (ulteriore), col fine di favorire l'erogazione del resto

OUTPUT:richiesta_denaro_utente_per_resto

if DETECT:annulla_operazione

then

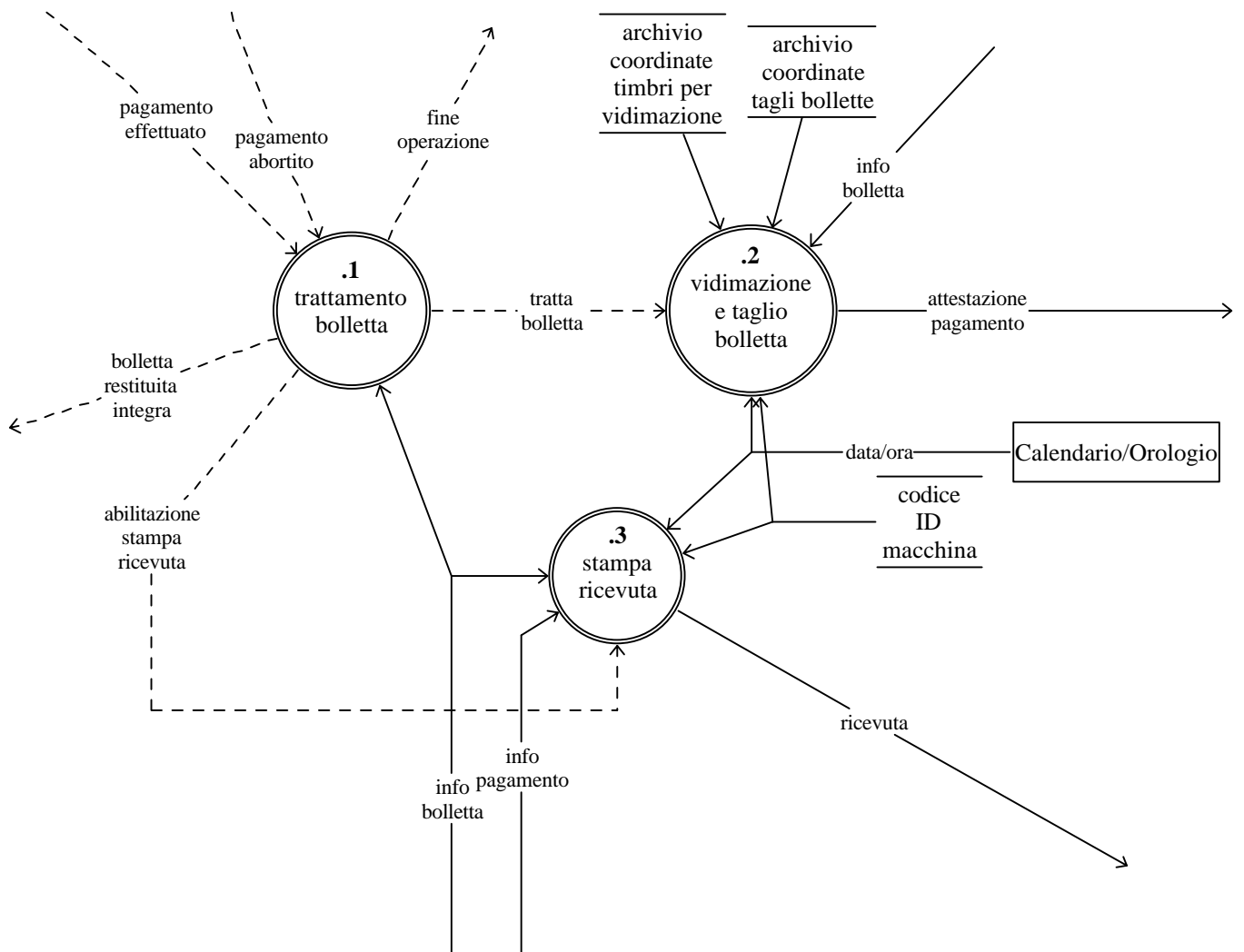
EMIT:restituisce_denaro_inserito

OUTPUT:info_resto

if DETECT:conferma

then

PUT:massimo_resto_da_fornire = GET:massimo_resto_emissibile



if DETECT:tratta_bolletta

then

INPUT:info_bolletta

READ:codice_ID_macchina

GET:data/ora

READ:archivio_coordinate_tagli_bollette

READ:archivio_coordinate_timbri_per_vidimazione

In base ai dati degli archivi (in relazione al tipo della bolletta), fornisce le informazioni necessarie al taglio ed alla vidimazione della bolletta fisica

OUTPUT:attestazione_pagamento

```
if INPUT:info_bolletta . tipo_bolletta = unknown or
  DETECT:pagamento_abortito = TRUE
then
  EMIT:bolletta_restituita_integra = TRUE

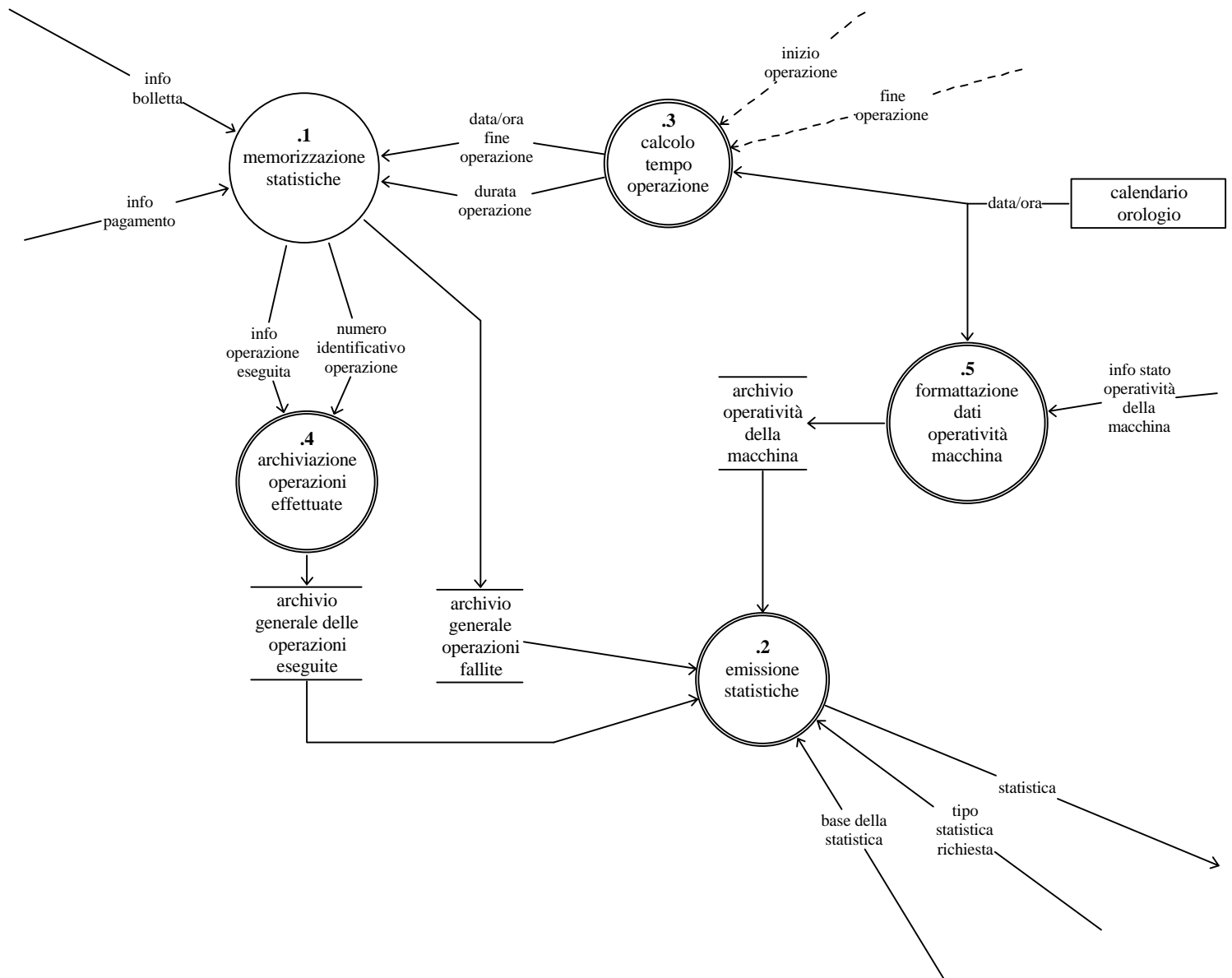
if DETECT:pagamento_effettuato = TRUE and
  not ( INPUT:info_bolletta . tipo_bolletta = unknown )
then
  USE:info_bolletta
  EMIT:tratta_bolletta
  EMIT:abilitazione_stampa_ricevuta

EMIT:fine_operazione
```

```
if DETECT:abilitazione_stampa_ricevuta
then
  INPUT:info_bolletta
  INPUT:info_pagamento
  READ:codice_ID_macchina
  GET:data/ora
```

In base ai dati letti, fornisce le informazioni per la stampa della ricevuta

```
OUTPUT:ricevuta
```



INPUT:info_stato_operatività_della_macchina

INPUT:data/ora

WRITE:archivio_operatività_della_macchina

```
WRITE:archivio_generale_delle_operazioni_eseguite =  
( GET:numero_identificativo_operazione ,  
  GET:info_operazione_eseguita
```

INPUT:tipo_statistica_richiesta

INPUT:base_della_statistica

READ:archivio_generale_delle_operazioni_eseguite

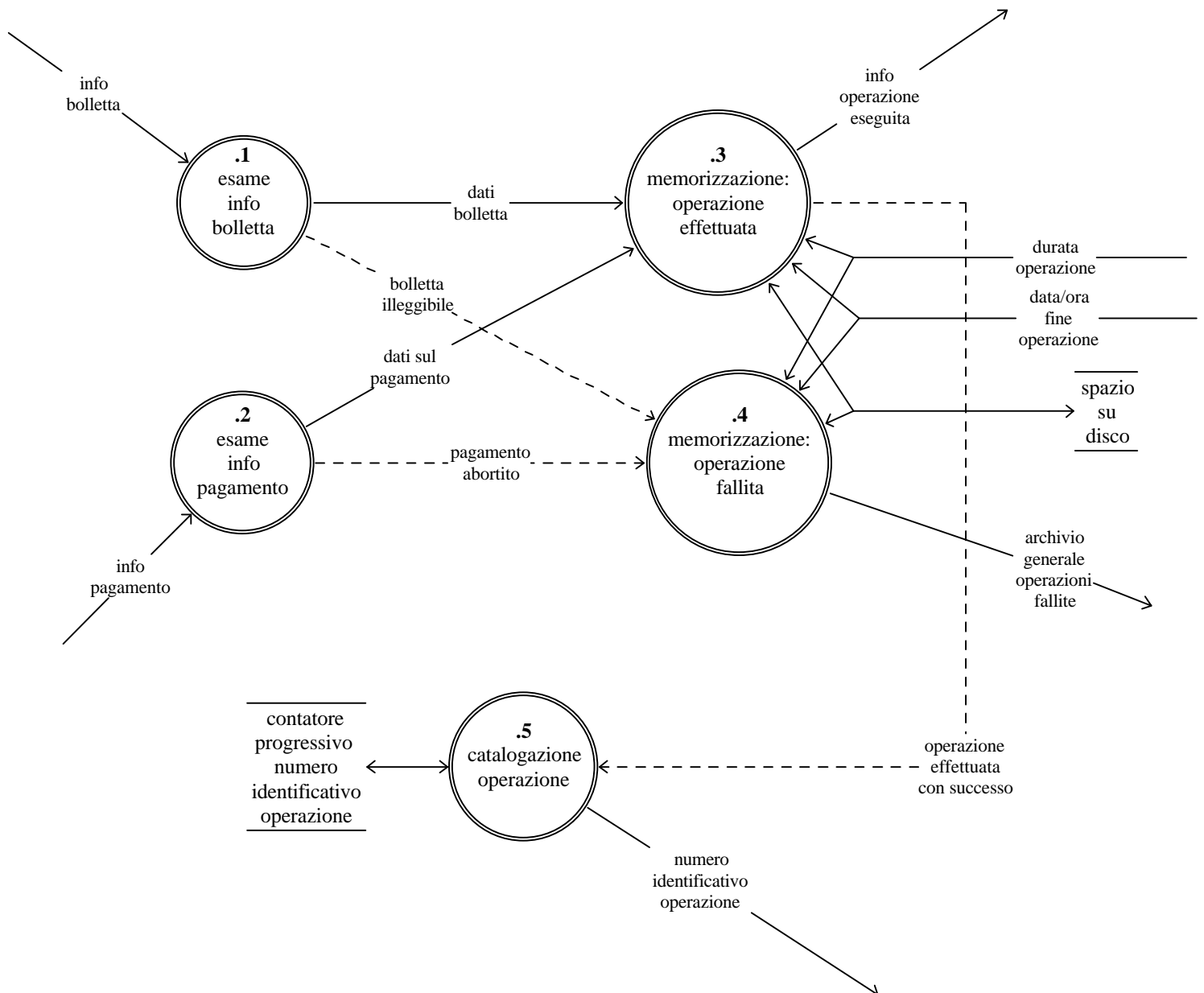
READ:archivio_operatività_della_macchina

READ:archivio_generale_operazioni_fallite

OUTPUT:statistica

```
if DETECT:inizio_operazione then start = <assoluto> ( GET:data/ora )  
if DETECT:fine_operazione then end = <assoluto> ( GET:data/ora )
```

```
OUTPUT:durata_operazione = end - start  
OUTPUT:data/ora_fine_operazione = end
```

```
if RECOGNIZE:pagamento_abortito = TRUE
then OUTPUT:archivio_generale_operazioni_fallite =
[ INPUT:data/ora_fine_operazione ,
  INPUT:durata_operazione
  Abort Utente
```

```
if RECOGNIZE:bolletta_illeggibile = TRUE
then OUTPUT:archivio_generale_operazioni_fallite =
[ INPUT:data/ora_fine_operazione
  0
  Bolletta Illegibile
```

```
UPDATE:spazio_su_disco
```

INPUT: durata_operazione

INPUT: dati_sul_pagamento

INPUT: dati_bolletta

INPUT: data/ora_fine_operazione

GENERATE: operazione_effettuata_con_successo

OUTPUT: info_operazione_eseguita

UPDATE: spazio_su_disco

USE:info_bolletta

GENERATE:bolletta_illeggibile

OUTPUT:dati_bolletta

INPUT:info_pagamento

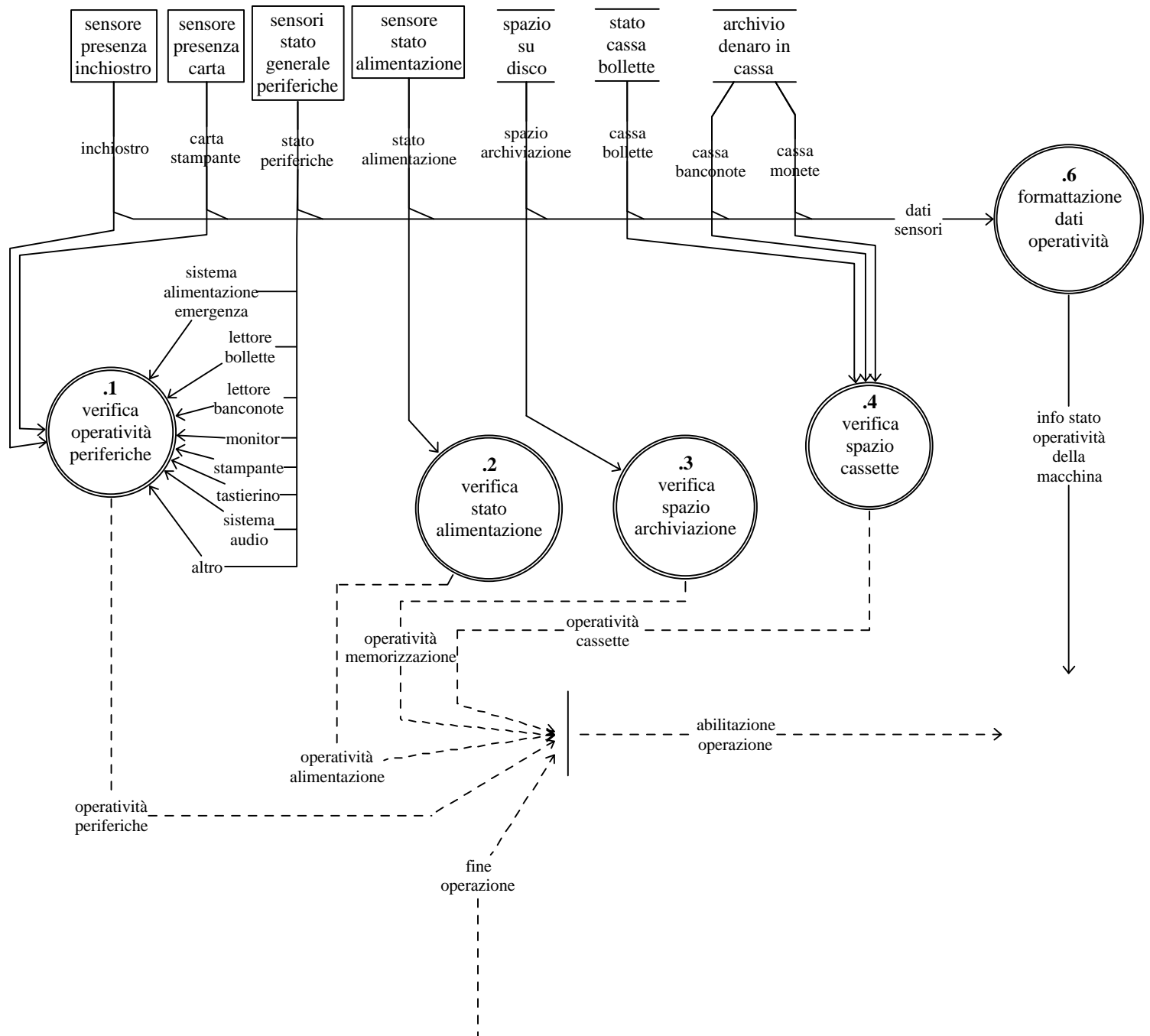
GENERATE:pagamento_abortito

OUTPUT:dati_sul_pagamento

if RECOGNIZE:operazione_effettuata_con_successo = TRUE

then OUTPUT:numero_identificativo_operazione = READ:contatore_progressivo__numero_identificativo_op

UPDATE:contatore_progressivo__numero_identificativo_operazione



GET:dati_sensori

OUTPUT:info_stato_operatività_della_macchina


```
if GET:cassa_bollette > 95 or  
   GET:cassa_banconote > 95 or  
   GET:cassa_monete > 95
```

```
then GENERATE:operatività_cassette = NON OPERATIVO  
else GENERATE:operatività_cassette = OPERATIVO
```

if GET:spazio_archiviazione < 3

then GENERATE:operatività_memorizzazione = NON OPERATIVO

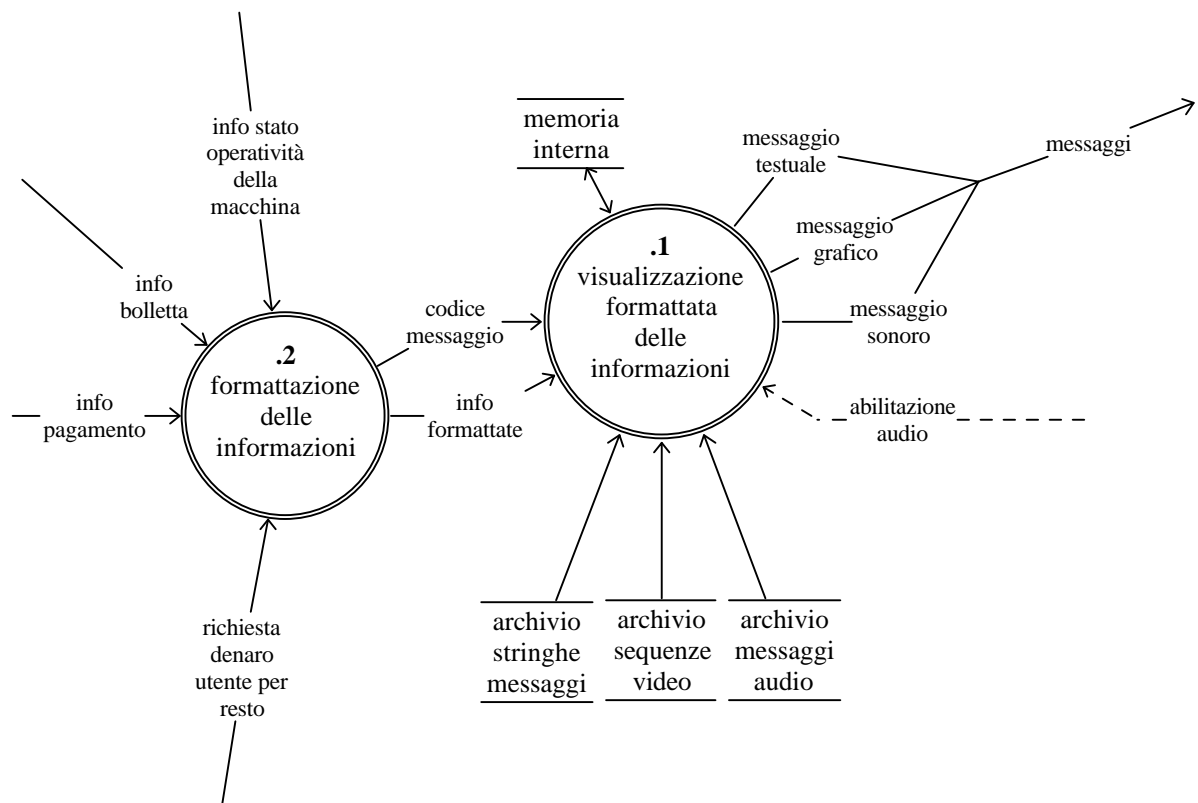
else GENERATE:operatività_memorizzazione = OPERATIVO

```
if GET:stato_alimentazione = ( alimentazione_corretta = TRUE )  
  
then GENERATE:operatività_alimentazione = OPERATIVO  
else GENERATE:operatività_alimentazione = NON OPERATIVO
```

```
if GET:inchostro > 3 and
if GET:carta_stampante > 3 and
if GET:tastierino = FUNZIONANTE and
if GET:monitor = FUNZIONANTE and
if GET:sistema_audio = FUNZIONANTE and
if GET:sistema_alimentazione_emergenza = FUNZIONANTE and
if GET:stampante = FUNZIONANTE and
if GET:lettore_banconote = FUNZIONANTE and
if GET:lettore_bollette = FUNZIONANTE and
if GET:altro = FUNZIONANTE

then GENERATE:operatività_periferiche = OPERATIVO
else GENERATE:operatività_periferiche = NON OPERATIVO
```

Input					Output
operatività periferiche	operatività memorizzazione	operatività cassette	fine operazione	operatività alimentazione	abilitazione operazione
OPERATIVO	OPERATIVO	OPERATIVO	TRUE	OPERATIVO	ABILITATA
OTHERS					DISABILITATA



Memorizzazione temporanea, in un buffer interno, delle informazioni da produrre in uscita

WRITE:memoria_interna = GET:info_formattate

GET:codice_messaggio

READ:archivio_sequenze_video

READ:archivio_messaggi_audio

READ:archivio_stringhe_messaggi

USE:memoria_interna

Si usa il codice del messaggio come chiave per gli archivi dei messaggi, che vengono quindi individuati e forniti all'utente

if RECOGNIZE:abilitazione_audio = TRUE

then OUTPUT:messaggio_sonoro

OUTPUT:messaggio_testuale

OUTPUT:messaggio_grafico

INPUT:richiesta_denaro_utente_per_resto
INPUT:info_stato_operatività_della_macchina
INPUT:info_pagamento
INPUT:info_bolletta

Formatta i dati che giungono dai vari moduli, classificandoli mediante un codice alfanumerico cui si associa un campo informativo (con i dati ricevuti).

PUT:codice_messaggio
PUT:info_formattate

Name	Definition	Type
abilitazione audio	*abilitazione della lettura vocale dei messaggi testuali (di supporto agli utenti con difficoltà visive)* DOMAIN: [TRUE FALSE]	Control
abilitazione lettura	DOMAIN: [TRUE FALSE]	Control
abilitazione operazione	*abilitazione dell'operazione* DOMAIN: [ABILITATA DISABILITATA]	Control
abilitazione stampa ricevuta	DOMAIN: [TRUE FALSE]	Control
alimentazione corretta	*corretta alimentazione elettrica della macchina* DOMAIN: [TRUE FALSE]	
alimentazione di emergenza	autonomia_emergenza *alimentazione elettrica della macchina mediante batterie di emergenza*	
altro	*stato funzionamento complessivo del sistema (eventuali malfunzionamenti non classificati)* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
anno	[1999..2999]	
annulla	DOMAIN: [TRUE FALSE]	Control
annulla operazione	*Operazione abortita per annullamento da parte dell'utente (dall'esterno) o perchè scade il tempo in cui è necessario terminare l'operazione* DOMAIN: [TRUE FALSE]	Control
annullamento operazione	*permette all'utente di annullare l'operazione in esecuzione* DOMAIN: [TRUE FALSE]	Control
archivio codici a barre	{[tipo_codice_a_barre tipo_decodifica_codice_a_barre]} *informazioni sulla decodifica dei codici a barre estratti dalla bolletta*	Data
archivio coordinate tagli bollette	1 {[tipo_bolletta + spostamento_taglio]}	Data
archivio coordinate timbri per vidimazione	1 {[tipo_bolletta + coordinate_timbro]}	Data
archivio costo operazione pagamento	*costo relativo all'effettuazione dell'operazione di pagamento automatico* DOMAIN: real	Data
archivio denaro in cassa	[1 {[quantità_banconote]}35 + 1 {[quantità_monete]}11] *elenco del denaro attualmente a disposizione nelle casse della macchina*	Data
archivio generale delle operazioni eseguite	{[numero_identificativo_operazione + data/ora_fine_operazione + durata_operazione + tipo_bolletta + intestatario_bolletta + importo_bolletta + importo_inserito + importo_reso]}	Data
archivio generale operazioni fallite	{[estremi_operazione_fallita]}	Data
archivio maschere fatture	{[immagine_fattura]}	Data
archivio messaggi audio	{[codice_messaggio + codice_messaggio_sonoro]}	Data
archivio operatività della macchina	{[data/ora_inizio_stato_macchina + codice_stato_operatività]}	Data
archivio sequenze video	{[codice_messaggio + codice_messaggio_video]}	Data
archivio stringhe messaggi	{[stringa_messaggio]}	Data

Name	Definition	Type
archivio timbri	*archivio delle timbrature*	Data
archivio valore denaro	[1 {[valore_banconota]}35 + 1 {[valore_moneta]}11] *mette in corrispondenza il tipo di banconota/moneta con il suo valore*	Data
arrivo nuova bolletta	DOMAIN: [TRUE FALSE]	Control
attestazione pagamento	[coordinate_timbro + spostamento_taglio + dati_timbro] *informazioni sul trattamento della bolletta*	Data
autonomia emergenza	secondi *tempo rimanente di autonomia nell'uso delle batterie per l'alimentazione di emergenza della macchina*	
banconota	[1..35] *codice classificazione banconota*	
base della statistica	*intervallo temporale sul quale effettuare estrazione di informazione dagli archivi della macchina*	Data
bolletta	[immagine_fattura] *immagine bolletta in ingresso*	Data
bolletta da trattare	DOMAIN: [TRUE FALSE]	Control
bolletta illeggibile	DOMAIN: [TRUE FALSE]	Control
bolletta restituita integra	*segnale di rilascio della bolletta (priva di vidimazioni e/o tagli), per annullamento operazione* DOMAIN: [TRUE FALSE]	Control
bytes	*bytes* DOMAIN: integer	
carta stampante	quantità_carta *quantità di carta rimanente (per rilasciare le ricevute del pagamento)*	Data
cassa banconote	riempimento_cassa *presenza di spazio per inserire ancora banconote *	Data
cassa bollette	riempimento_cassa *quantità di spazio per inserire ancora bollette*	Data
cassa monete	riempimento_cassa *presenza di spazio per inserire ancora monete*	Data
causa fallimento	*causa del fallimento di una operazione* DOMAIN: [Bolletta Illegibile Abort Utente Tempo Scaduto]	
codice ID macchina	id_macchina	Data
codice info	*identifica il tipo di informazione del pacchetto* DOMAIN: string	
codice messaggio	*codice identificativo del messaggio da inviare all'utente* DOMAIN: string	Data
codice messaggio sonoro	DOMAIN: string	
codice messaggio video	DOMAIN: string	
codice stato operatività	DOMAIN: [OPERATIVA End_Carta End_Ink End_Spazio_archiviazione Full_cassa_bollette Full_cassa_banconote Full_cassa_monete Allarme_alimentazione Allarme_periferiche]	
codice tipo bolletta	DOMAIN: string	

Name	Definition	Type
codice utente	*codice identificativo dell'utente, da considerare al posto del nominativo ed eventuali altre informazioni personale. Nel rispetto delle leggi sulla privacy.* DOMAIN: string	
conferma	*permette all'utente di confermare l'operazione in atto* DOMAIN: [TRUE FALSE]	Control
conferma/annulla	[conferma annulla_operazione] *pressione tasti 'conferma' o 'annulla'*	
contatore progressivo numero identificativo operazione	{ [numero_identificativo_operazione] }	Data
coordinate timbro	*coordinata x,y (orizzontale da sinistra, verticale dal basso) della posizione dove timbrare la bolletta* DOMAIN: [integer , integer]	
costo operazione	*imposta sull'utilizzo della macchina, quale contributo alle spese di gestione* DOMAIN: real	
data	[giorno_della_settimana+giorno_del_mese+mese+anno] *data*	
data-ora pagamento	[data + orario] *data ed orario di effettuazione del pagamento (fine operazione)*	
data/ora	[data + orario]	Data
data/ora fine operazione	[data + orario]	Data
data/ora inizio operazione	[data+orario] *data ed orario di inizio dell'operazione*	
data/ora inizio stato macchina	[data + orario] *data ed orario della modifica dello stato di operatività della macchina*	
dati bolletta	[tipo_bolletta + intestatario_bolletta + importo_bolletta]	Data
dati sensori	[stato_periferiche carta_stampante inchiostro spazio_archiviazione cassa_bollette cassa_banconote cassa_monete stato_alimentazione] *dati provenienti dai vari sensori della macchina *	Data
dati sul pagamento	[importo_inserito + importo_reso]	Data
dati timbro	[id_macchina + data-ora_pagamento + importo_bolletta + costo_operazione] *dati che saranno stampati sulla parte della bolletta da restituire all'utente*	
denaro	1 { [banconota moneta banconota + moneta] }	
denaro bolletta	denaro *denaro che l'utente inserisce per il pagamento della bolletta*	Data
denaro per resto	denaro *denaro che l'utente inserisce dopo suggerimento della macchina (ad importo raggiunto) per favorire l'emissione del resto.*	Data
denaro utente	[denaro_bolletta denaro_per_resto] *denaro immesso dall'utente*	Data
durata operazione	secondi	Data
E. attestazione pagamento	*bolletta tagliata e vidimata (l'operazione è stata conclusa con successo)*	Data
E. banconota accettata	*banconota (fisica) accettata (dopo i controlli) dalla macchina*	Data
E. banconota immessa	*banconota (fisica) immessa nella macchina*	Data
E. banconote depositate	*banconote (fisiche) depositate nella cassa-banconote*	Data
E. banconote tmp	*banconote (fisiche) conservate temporaneamente*	Data
E. bolletta accettata	*bolletta (fisica) inserita nella macchina ed accettata da questa*	Data

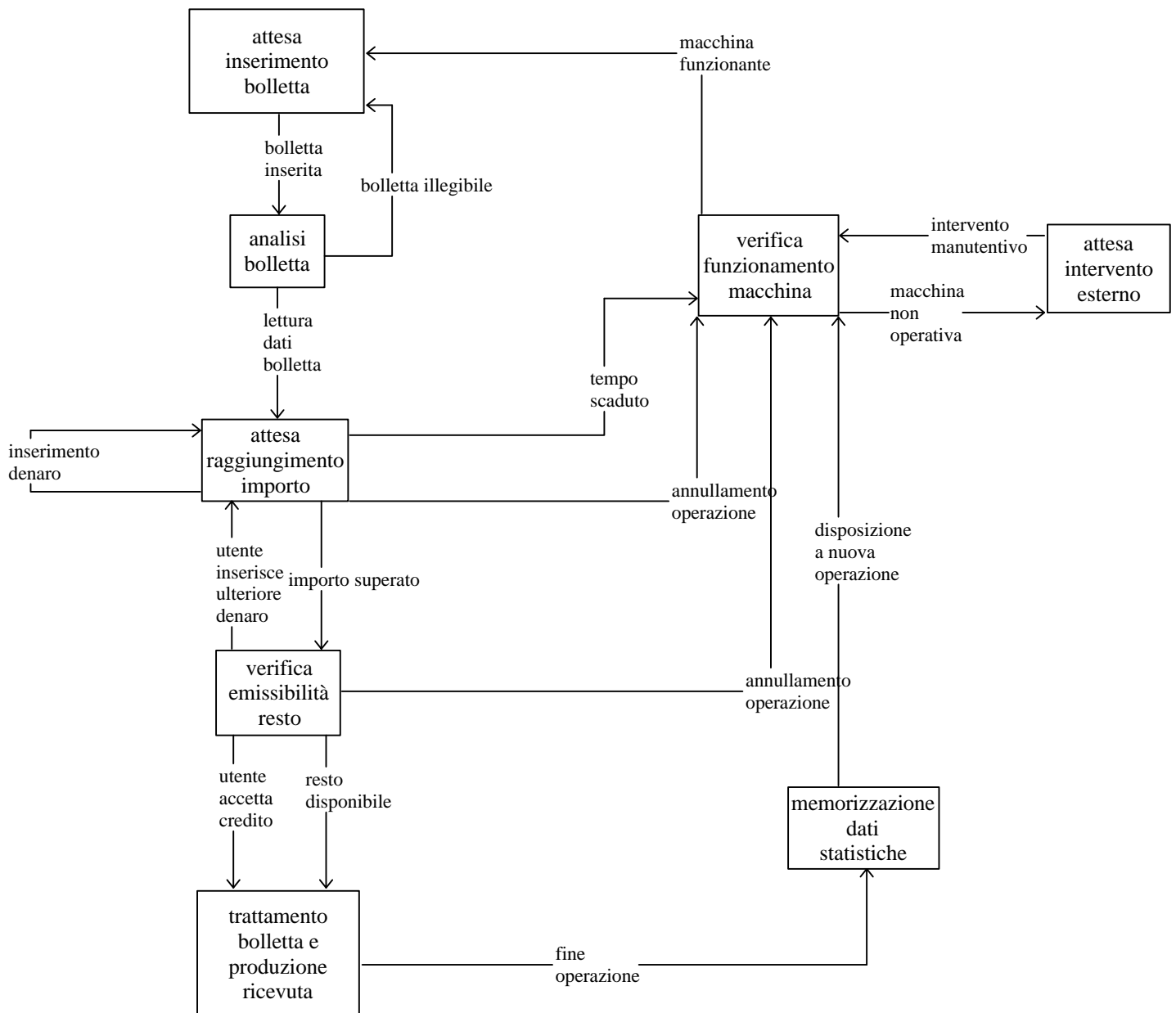
Name	Definition	Type
E. bolletta da trattare	*bolletta (fisica) accettata dalla macchina e diretta al modulo di trattamento (il pagamento è stato effettuato quindi con successo)*	Data
E. bolletta inserita	*bolletta (fisica) inserita nella macchina*	Data
E. bolletta restituita integra	*bolletta (fisica) restituita integra (priva di tagli o vidimazioni) dalla macchina. L'operazione è stata annullata dall'utente o la bolletta non viene considerata leggibile*	Data
E. denaro immesso	[0{[E._banconota_immessa]} + 0{[E._moneta_immessa]}] *denaro (fisico) inserito nella macchina*	Data
E. denaro restituito	*denaro (fisico) emesso dalla macchina*	Data
E. moneta accettata	*moneta (fisica) accettata (dopo i controlli) dalla macchina*	Data
E. moneta immessa	*moneta (fisica) immessa nella macchina*	Data
E. monete depositate	*monete (fisiche) depositate nella cassa-monete*	Data
E. monete tmp	*monete (fisiche) conservate temporaneamente*	Data
E. output audio	*istruzioni ed informazioni mediante speaker*	Data
E. output grafico	*istruzioni ed informazioni mediante video*	Data
E. parte bolletta trattata	*parte della bolletta (fisica) che rimane nella macchina dopo l'operazione (terminata con successo)*	Data
E. pressione tasto	TBD	Data
E. ricevuta cartacea	*ricevuta (fisica) di pagamento effettuato*	Data
E. richiesta statistica	*statistica richiesta dall'operatore (attraverso un'interfaccia con un terminale o un palmare connesso alla macchina)*	Data
E. scelte utente	*scelte che l'utente effettua attraverso il tastierino (mediante pressione -fisica- dei tasti)*	
E. statistica fornita	*statistica fornita all'operatore (attraverso un'interfaccia con un terminale o un palmare connesso alla macchina)*	Data
eccedenza pagamento	*importo inserito in eccesso rispetto a quello dovuto* DOMAIN: real	Data
effettivo importo da pagare	DOMAIN: real	Data
estremi operazione fallita	[data/ora_inizio_operazione + durata_operazione + causa_fallimento]	
estremi operazioni effettuate	[numero_identificativo_operazione tipo_bolletta dati_bolletta data/ora_inizio_operazione durata_operazione] *estremi delle operazioni effettuate con successo*	
fine operazione	*indica la terminazione dell'operazione* DOMAIN: [TRUE FALSE]	Control
fine operazione lettura	*fine operazione di lettura. E' il segnale di inizio operazione di pagamento (start timer,...)* DOMAIN: [TRUE FALSE]	Control
giorno del mese	[1..31]	
giorno della settimana	DOMAIN: [Lun Mar Mer Gio Ven Sab Dom]	
id macchina	*codice alfanumerico identificativo della macchina* DOMAIN: string	
immagine fattura	DOMAIN: [immagine]	
importo bolletta	DOMAIN: real	Data

Name	Definition	Type
importo da rendere	*importo che non è stato possibile fornire all'utente come resto. Rappresenta un credito dell'utente, che può riavere il denaro dovuto presentando la ricevuta ad un operatore (umano) in orario d'ufficio.* DOMAIN: real	
importo inserito	*valore del denaro inserito dall'utente per il pagamento della bolletta* DOMAIN: real	
importo inserito per resto	*importo inserito dall'utente per semplificare (o permettere) le operazioni di fornitura del resto, su suggerimento della stessa macchina* DOMAIN: real	
importo reso	*valore del denaro fornito all'utente come resto dell'operazione di pagamento* DOMAIN: real	Data
importo richiesto per resto	DOMAIN: real	
inchiostro	percentuale *quantità di inchiostro rimanente per timbri e stampa ricevute*	Data
info bolletta	[tipo_bolletta + intestatario_bolletta + importo_bolletta] *informazioni associabili alla bolletta letta dalla macchina*	Data
info formattate	[tipo_bolletta intestatario_bolletta importo_bolletta effettivo_importo_da_pagare importo_inserito importo_richiesto_per_resto importo_inserito_per_resto importo_reso codice_stato_operatività] *informazioni associate ai messaggi per l'utente*	Data
info operazione eseguita	[data/ora_fine_operazione + durata_operazione + tipo_bolletta + intestatario_bolletta + importo_bolletta + importo_inserito + importo_reso]	Data
info pagamento	[codice_info + stato_pagamento + effettivo_importo_da_pagare + importo_inserito + importo_richiesto_per_resto + importo_inserito_per_resto + importo_reso] *informazioni associabili all'effettuazione del pagamento*	Data
info resto	[importo_richiesto_per_resto + importo_inserito_per_resto + importo_reso]	Data
info stato operatività della macchina	[codice_info + codice_stato_operatività] *informazioni sullo stato attuale di operatività della macchina*	Data
informazioni stato macchina	[data/ora_inizio_stato_macchina + codice_stato_operatività] *informazioni sullo stato (operabilità) della macchina*	
inizio operazione	*indica l'inizio dell'operazione* DOMAIN: [TRUE FALSE]	Control
intestatario bolletta	codice_utente	
lettore banconote	*stato funzionamento del lettore delle banconote* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
lettore bollette	*stato funzionamento del lettore delle bollette utente* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
massimo resto da fornire	*resto che la macchina è in grado di fornire all'utente (approssimazione per difetto del resto dovuto ma non fornibile)* DOMAIN: real	Data
massimo resto emissibile	*massimo resto (inferiore a quello dovuto) che la macchina ha possibilità di fornire all'utente* DOMAIN: real	Data

Name	Definition	Type
memoria interna	*memoria interna dedicata al processo per la memorizzazione temporanea di dati in ingresso o in uscita* DOMAIN: [string integer real]	Data
mese	DOMAIN: [Gen Feb Mar Apr Mag Giu Lug Ago Set Ott Nov Dic]	
messaggi	[messaggio_testuale + messaggio_grafico + messaggio_sonoro] *istruzioni per l'utente e messaggi generi all'utenza*	Data
messaggi audio	*archivio dei messaggi audio preregistrati*	Data
messaggio grafico	codice_messaggio_video *sequenza di immagini, esemplificativa dell'operazione da svolgere, quale valido aiuto all'utenza (per la sua immediata comprensibilità)*	Data
messaggio sonoro	codice_messaggio_sonoro *istruzioni vocali all'utente, quale guida per le persone con scarsa dimestichezza con terminali video o con difficoltà nella lettura da video.*	Data
messaggio testuale	*messaggio sotto forma di stringhe visualizzate sullo schermo-display* DOMAIN: string	Data
min	[0..59]	
moneta	[0..11] *codice classificazione moneta*	
monitor	*stato del display* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
numero attuale bollette	DOMAIN: integer	
numero identificativo operazione	*numero progressivo che identifica univocamente l'operazione eseguita* DOMAIN: integer	Data
numero max bollette	DOMAIN: integer	
numero operazione	*numero progressivo identificativo dell'operazione effettuata* DOMAIN: integer	
numero operazioni effettuate	*numero delle operazioni che si sono concluse con successo* DOMAIN: integer	
operatività alimentazione	*operatività sistema alimentazione* DOMAIN: [OPERATIVO NON OPERATIVO]	Control
operatività cassette	*operatività complessiva delle casse* DOMAIN: [OPERATIVO NON OPERATIVO]	Control
operatività memorizzazione	*operatività complessiva sistema di memorizzazione* DOMAIN: [OPERATIVO NON OPERATIVO]	Control
operatività periferiche	*complessiva operatività periferiche* DOMAIN: [OPERATIVO NON OPERATIVO]	Control
operazione effettuata con successo	DOMAIN: [TRUE FALSE]	Control
ora	[0..23]	
orario	[ora+min+sec]	Data
pagamento abortito	DOMAIN: [TRUE FALSE]	Control

Name	Definition	Type
pagamento completato	DOMAIN: [TRUE FALSE]	Control
pagamento effettuato	DOMAIN: [TRUE FALSE]	Control
parametri accettazione banconote	*archivio contenente i parametri da verificare per l'accettazione delle banconote*	Data
parametri accettazione monete	*archivio contenente i parametri da verificare per l'accettazione delle monete*	Data
parametri di accettazione bollette	*parametri utilizzati dal modulo di accettazione delle bollette*	Data
percentuale	[0 .. 100]	
quantità banconote	*quantità banconote presente nell i-mo cassetto (banconote di valore i-mo)* DOMAIN: integer	
quantità carta	percentuale	
quantità monete	*quantità di monete presente nel tubo j-mo (di valore j-mo)* DOMAIN: integer	
restante spazio memorizzazione	bytes *spazio ancora utilizzabile sul supporto di memorizzazione per l'archiviazione delle statistiche*	
restituisce denaro inserito	*(a seguito di annullamento dell'operazione)* DOMAIN: [TRUE FALSE]	Control
resto da fornire	*importo del resto che la macchina deve (e può) fornire* DOMAIN: real	Data
ricevuta	[numero_operazione + id_macchina + data-ora_pagamento + tipo_bolletta + importo_bolletta + costo_operazione + denaro_utente + importo_reso + importo_da_rendere] *informazioni da stampare sulla ricevuta del pagamento effettuato*	Data
richiesta denaro utente per resto	[codice_info + importo_richiesto_per_resto] *richiesta introduzione denaro in eccesso, per semplificazione delle operazioni di emissione del resto*	Data
riempimento cassa	percentuale *percentuale riempimento della cassa*	
sec	[0..59]	
secondi	*secondi* DOMAIN: integer	
sistema alimentazione emergenza	*stato operatività delle batterie per l'alimentazione di emergenza* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
sistema audio	*stato del sistema audio* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
spazio archiviazione	restante_spazio_memorizzazione *spazio su disco fisso per l'archiviazione dei dati (statistiche varie)*	Data
spazio su disco	restante_spazio_memorizzazione *livello di riempimento del supporto di memorizzazione*	Data
spostamento taglio	*distanza dal bordo sinistro della bolletta, alla quale effettuare il taglio* DOMAIN: integer	

Name	Definition	Type
stampante	*stato funzionamento della stampante* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
statistica	[numero_operazioni_effettuate estremi_operazioni_effettuate tempo_medio_operazioni_effettuate estremi_operazione_fallita tempo_medio_operazioni_abortite informazioni_stato_macchina]	Data
stato alimentazione	[alimentazione_corretta alimentazione_di_emergenza] *stato alimentazione elettrica della macchina*	Data
stato cassa bollette	[numero_max_bollette numero_attuale_bollette] *livello di riempimento della cassa delle bollette trattate*	Data
stato pagamento	DOMAIN: [IN ATTO COMPLETATO ABORTITO TEMPO SCADUTO]	
stato periferiche	[tastierino + monitor + sistema_audio + sistema_alimentazione_emergenza + stampante + lettore_banconote + lettore_bollette + altro] *stato di funzionalità delle periferiche della macchina (video, stampante, tastierino, ...)*	Data
stringa messaggio	DOMAIN: string	
superamento tempo massimo	DOMAIN: [TRUE FALSE]	Control
tastierino	*stato del tastierino con pulsanti* DOMAIN: [FUNZIONANTE NON FUNZIONANTE]	Data
tempo massimo operazione di pagamento	*(in secondi)* DOMAIN: integer	Data
tempo medio operazioni abortite	secondi *tempo medio della durata delle operazioni effettuate senza successo*	
tempo medio operazioni effettuate	secondi *tempo medio della durata delle operazioni effettuate con successo*	
tipo bolletta	[codice_info + codice_tipo_bolletta] *tipo di fattura inserita (pagabile con la macchina)*	Data
tipo codice a barre	*modello identificativo del codice a barre* DOMAIN: string	
tipo decodifica codice a barre	*caratteristiche di decodifica del codice a barre* DOMAIN: string	
tipo statistica richiesta	*tipologia della statistiche che si richiede alla macchina* DOMAIN: [NUMERO OPERAZIONI EFFETTUATE ESTREMI OPERAZIONI EFFETTUATE TEMPO MEDIO OPERAZIONI EFFETTUATE ESTREMI OPERAZIONI FALLITE TEMPO MEDIO OPERAZIONI FALLITE INFORMAZIONI STATO MACCHINA]	Data
tratta bolletta	DOMAIN: [TRUE FALSE]	Control
valore banconota	*valore della banconota classificata di tipo i-mo* DOMAIN: real	
valore moneta	*valore della moneta classificata di tipo j-mo* DOMAIN: real	



Parte Quarta

**PROGETTAZIONE
DEL
SOFTWARE**

Software Design

Modularizzazione del progetto

Al fine di rendere più trattabile il progetto, al livello più astratto viene effettuata una suddivisione in moduli logici (suddivisione basata sull'affinità di talune operazioni e di alcuni processi con i relativi dispositivi hardware).

La creazione di moduli differenti, quasi del tutto indipendenti tra loro se non per opportuni interfacciamenti, rende il progetto sicuramente più chiaro e più leggibile. Una tale operazione si rivela estremamente utile in special modo per quel che riguarda l'approccio architetturale (trattato nel seguito).

I moduli che si è ritenuto utile individuare nel progetto (ci si basa sul diagramma Enhanced-DFD) sono elencati e sommariamente descritti di seguito:

1. **Modulo Interfacciamento in Input**

Modulo comprendente il software di gestione dei pulsanti presenti sulla macchina, con i quali l'utente può effettuare delle scelte (tasti: conferma, annulla ed abilitazione audio).

2. **Modulo Interfacciamento in Output**

Modulo che gestisce l'Output verso l'utente, sia come messaggi ed istruzioni visuali (testuali e grafiche) che come supporto audio.

3. **Modulo Gestione Bollette**

Modulo che racchiude tutti gli oggetti e le funzionalità (oltre che ai drivers dei dispositivi fisici) relative alla gestione delle bollette (dall'estrazione dei dati, alla preparazione della ricevuta).

4. **Modulo Gestione Denaro**

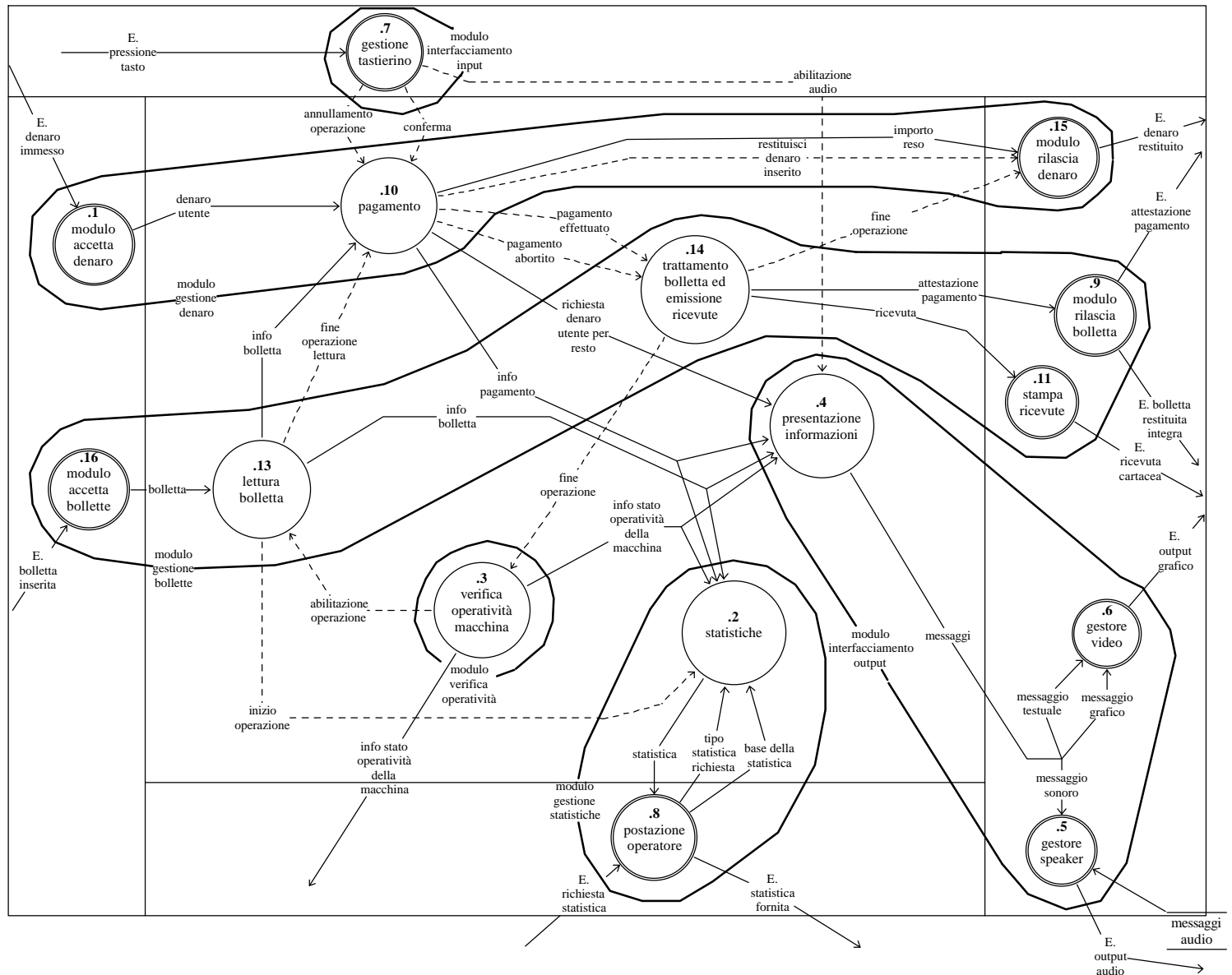
Modulo che identifica tutti i sottomoduli che trattano la gestione del denaro (dal conteggio di quello inserito dall'utente, alla complessa gestione del resto). Anche questo modulo contiene drivers di gestione delle periferiche hardware (quale, ad esempio, il riconoscitore di banconote).

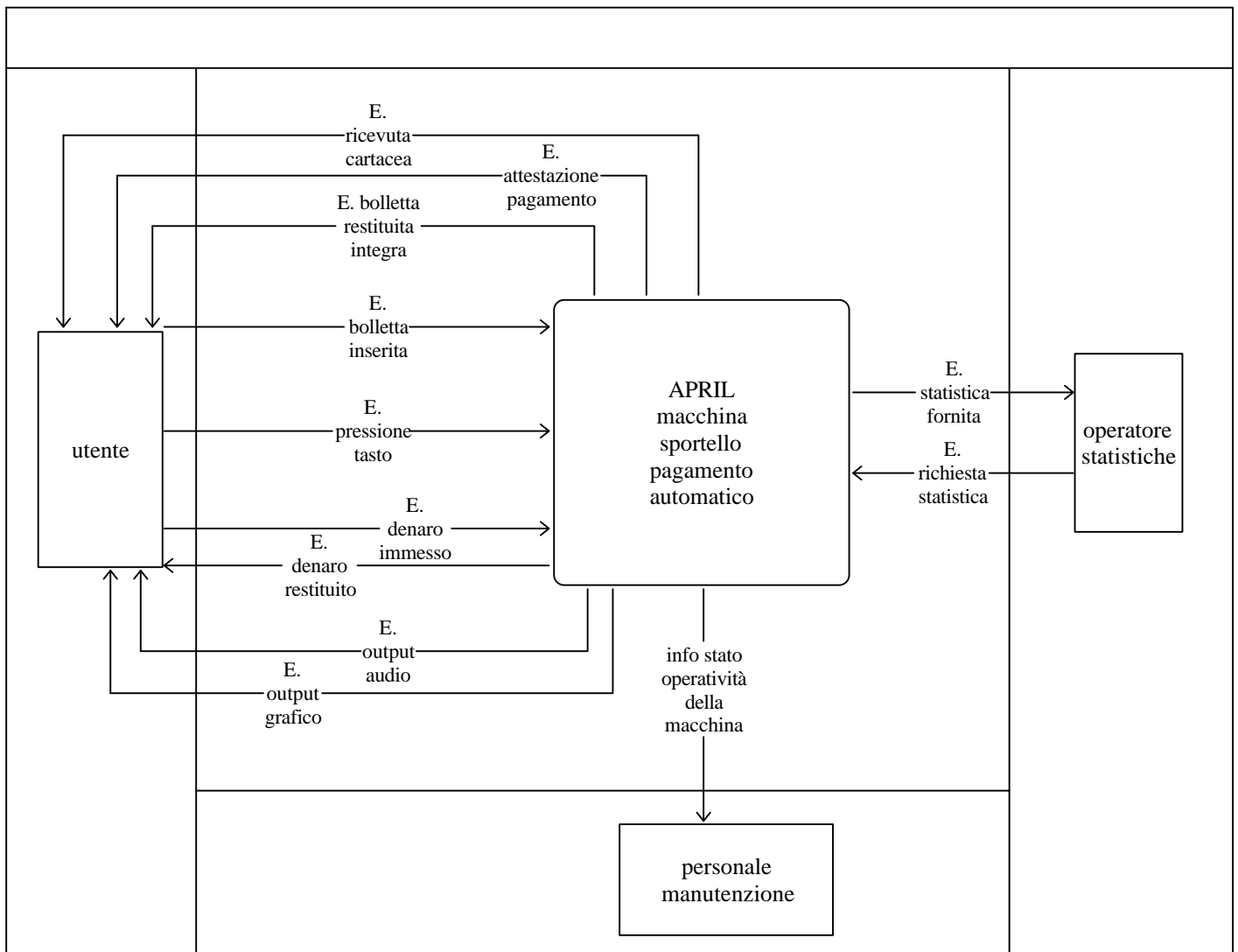
5. **Modulo Gestione Statistiche**

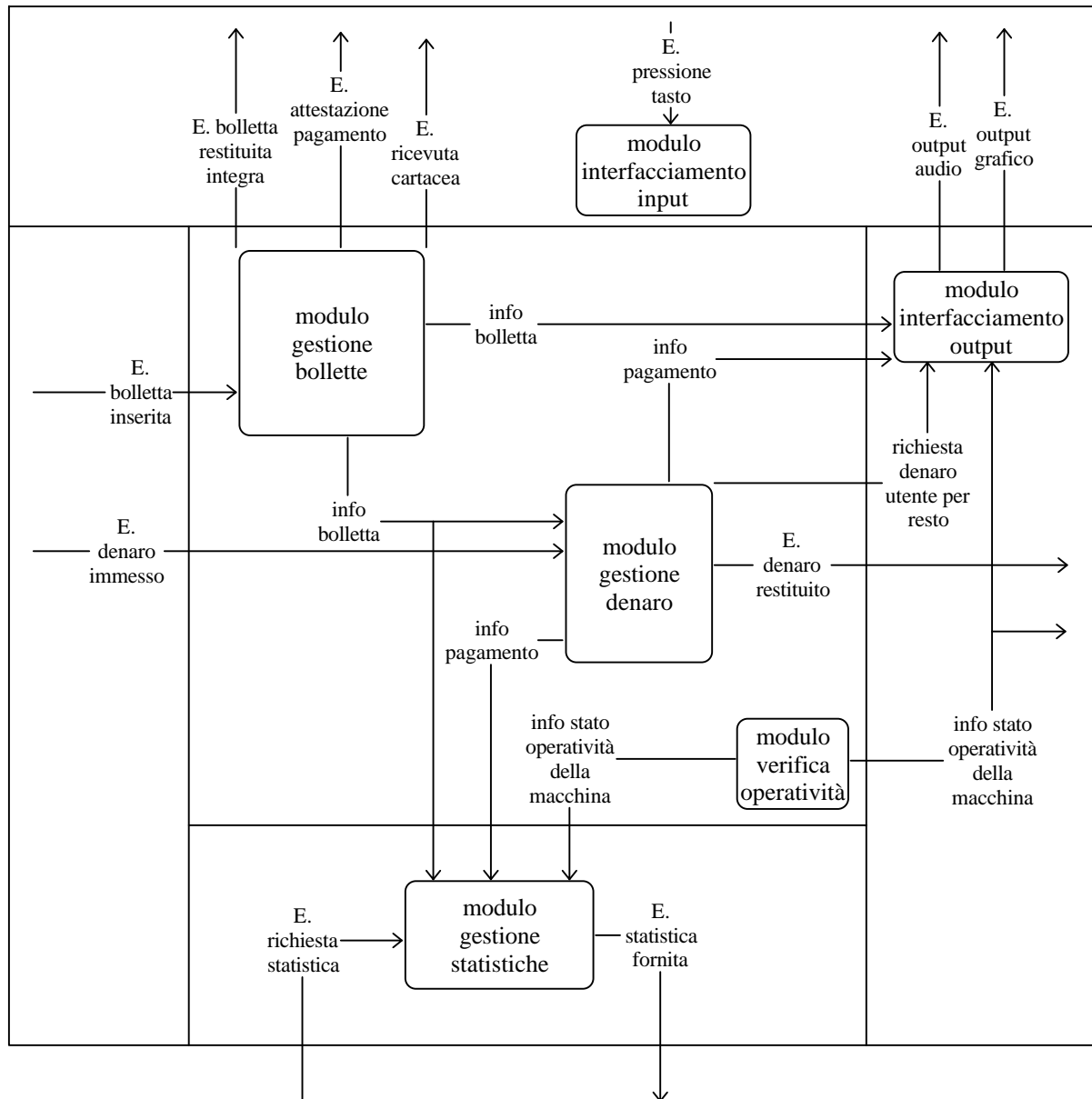
Modulo che consente la gestione completa delle statistiche (dalla memorizzazione dei dati raccolti durante l'operazione, alla fornitura delle statistiche richieste dall'operatore).

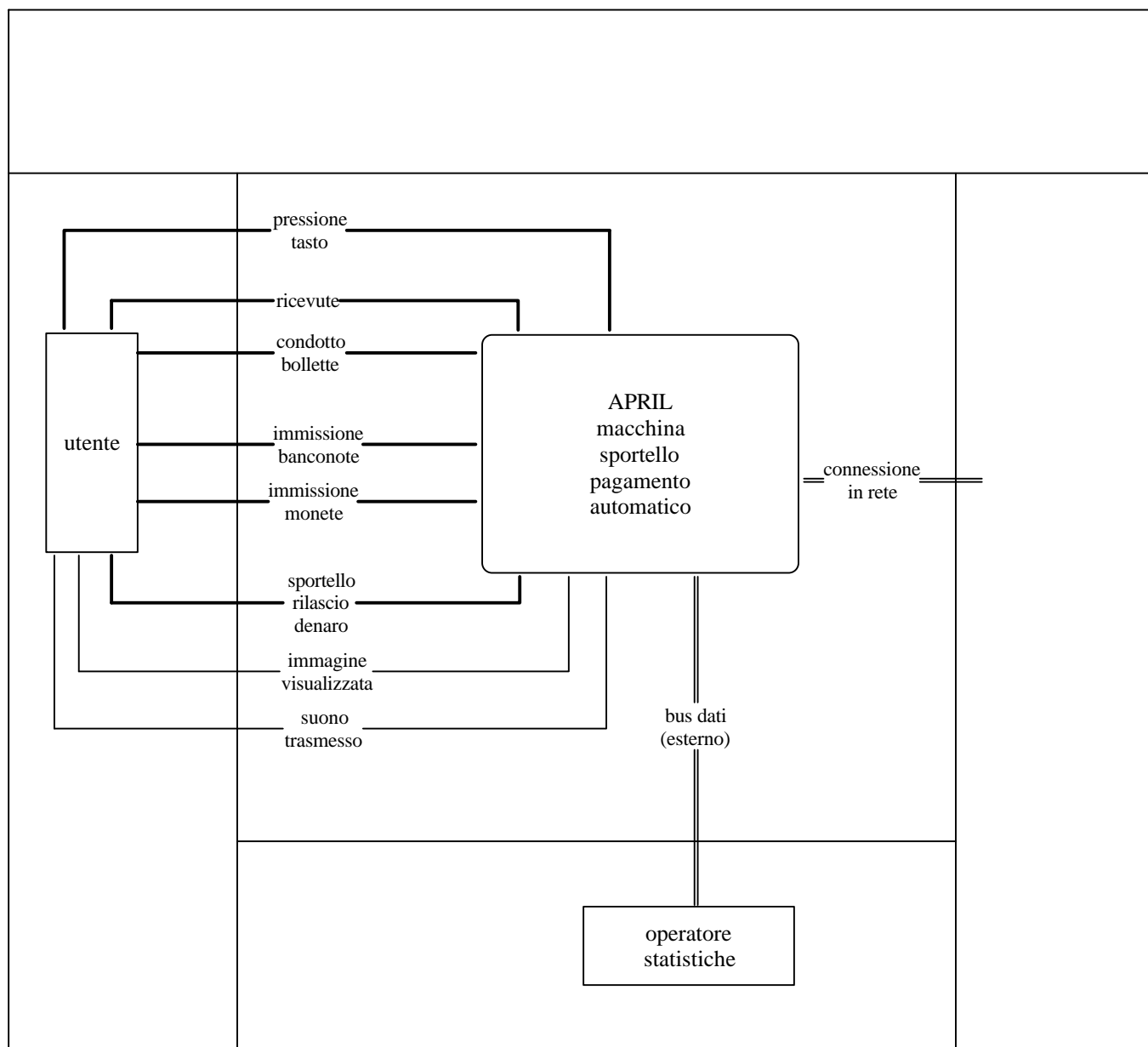
6. **Modulo Verifica Operatività della Macchina**

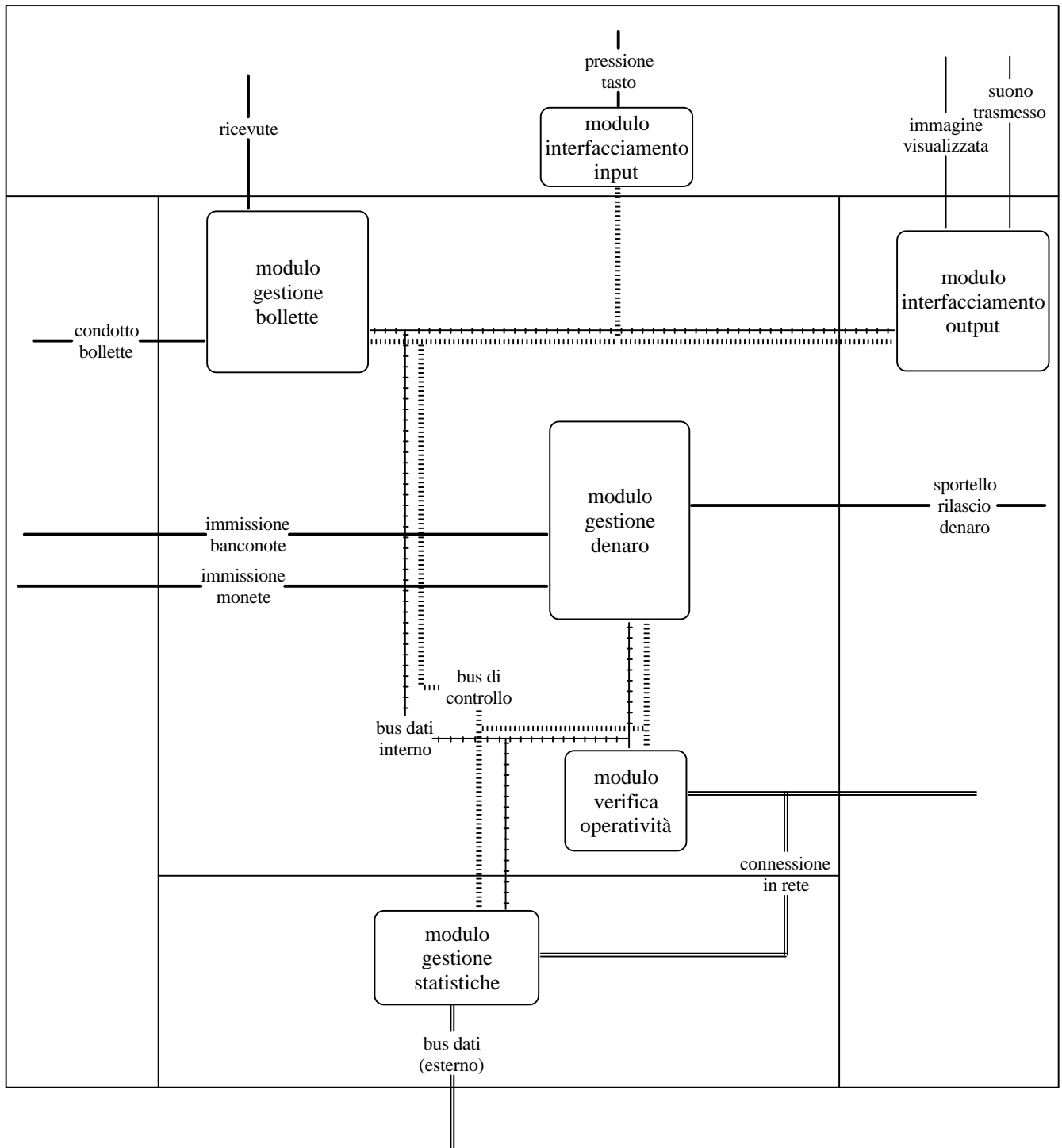
Modulo che gestisce il monitoraggio dell'intera macchina in modo da garantire la corretta terminazione delle operazioni iniziate (blocca l'utilizzo della macchina prima della successiva operazione, in caso di problemi). La diagnosi della macchina passa attraverso il testing dei vari dispositivi hardware ed il controllo di taluni registri.











Modularizzazione del software

Scelta del linguaggio di codifica

La tecnica di programmazione ritenuta più idonea è sicuramente quella orientata agli oggetti. Si tratta di una filosofia, di un modo di pensare ed immaginare il problema, più che di una vera e propria strategia di codifica.

La programmazione orientata agli oggetti (la cui diffusione è sicuramente molto ampia) riveste un ruolo fondamentale anche come autentico supporto alla concretizzazione di concetti teorici di grande importanza: ci si riferisce, ad esempio, all'information hiding, alla modularizzazione (che permette, tra le altre cose, una suddivisione efficace del lavoro ed una manutenibilità più semplice), alla riusabilità ed all'ereditarietà (tutti concetti del tutto familiari all'interno della programmazione *object oriented*, ma difficili da realizzare nei comuni linguaggi procedurali).

Posto questo, bisogna effettuare la scelta del linguaggio di programmazione.

Essenzialmente il problema ricade sulla scelta tra il linguaggio C++ ed il Java. Per il progetto presentato si è scelto il C++, a discapito del (forse più quotato) concorrente.

Le motivazioni di questa scelta sono essenzialmente concentrate sulla maggiore potenza del linguaggio C++. Il Java risulta più portabile, più orientato alle applicazioni grafiche, particolarmente semplice nella gestione del multithreading ma forse più limitato negli aspetti di efficienza e di ottimizzazione del prodotto finale. Di contro, il linguaggio scelto può essere considerato molto potente, flessibile, efficace ed efficiente. Svantaggio principale del C++ potrebbe forse essere l'eccessiva libertà data al programmatore che, unita alla potenza già accennata, potrebbe diventare pericolosa; è pur vero, però, che questo risulta un problema reale solamente per programmatori inesperti o con mentalità poco affini alle caratteristiche del linguaggio stesso; né l'una né l'altra cosa sono attribuibili ai programmatori del team di sviluppo del progetto a p r i l

MODULE Bolletta
USES

EXPORTS

data Ditta

Identifica l'azienda fornitrice del servizio fatturato

Il dato viene rilevato mediante connessione indiretta con il dispositivo periferico di riconoscimento delle fatture (per mezzo di *system calls*).

data Identificatore

Identifica la fattura tra quelle della stessa ditta (es. periodo di conteggio scatti telefonici, periodo di consumo acqua,...)

Il dato viene rilevato mediante connessione indiretta con il dispositivo periferico di riconoscimento delle fatture (per mezzo di *system calls*).

data CodCliente

Codice del Cliente (Nel trattamento della fattura ci si riferisce al cliente esclusivamente con il suo codice identificativo, assegnato dalla società di servizi che ha emesso la fattura).

Il dato viene rilevato mediante connessione indiretta con il dispositivo periferico di riconoscimento delle fatture (per mezzo di *system calls*).

data Importo

Importo fatturato (importo che l'utente è tenuto a pagare).

Il dato viene rilevato mediante connessione indiretta con il dispositivo periferico di riconoscimento delle fatture (per mezzo di *system calls*).

END exports

IMPLEMENTATION

Il modulo Bolletta rappresenta l'oggetto bolletta nella reale situazione di pagamento. Si osserva che i dati presentati nella parte pubblica sono tutti estratti dalla fattura mediante strumentazioni hardware (dispositivo di lettura della bolletta, decodificatore di barcode), comandate mediante *systems calls*, e sfruttando la tabella di decodifica dei barcode (considerata come parte privata).

END Bolletta

MODULE LetturaBolletta
USES Bolletta, Timer, Statistiche, Monitor

EXPORTS

fun ControllaOperatività ()

Effettua il test preliminare per stabilire l'operatività della macchina (e quindi permettere l'effettuazione di una nuova operazione, che inizia appunto con l'inserimento di una nuova bolletta).

Necessariamente si interfaccia al modulo dedicato alla verifica dell'operatività della macchina:

Monitor.Recognize (), che ritorna appunto l'abilitazione (o meno) alla nuova operazione.

fun IdentificaTipo ()

Identifica il tipo di bolletta inserita nel lettore.

Se la bolletta è leggibile:

1. Aziona il timer dell'intera operazione:
Timer.Start ()
2. Ritorna l'identificatore del tipo di bolletta, utilizzando l'oggetto istanziazione della classe Bolletta:
Bolletta.Ditta

Se la bolletta non è leggibile (o tipo *Unknown*):

1. Blocca il timer:
Timer.Reset ()
2. Memorizza le informazioni per le statistiche dei casi di fallimento:
GestoreStatistiche.NuovaOperazioneFallita (*Bolletta Illegibile*)
3. Permette l'espulsione della bolletta inserita (integra):
GestoreBollette.RilascioBolletta ()

fun EstraiDati ()

Estrae i dati della bolletta

Ritorna le informazioni ricavabili dalla bolletta:

Bolletta.CodCliente +
 Bolletta.Identificatore +
 Bolletta.Importo

END exports**IMPLEMENTATION**

Questo modulo coordina le operazioni di lettura della bolletta. Per accedere a determinate informazioni sfrutta il modulo Bolletta (che utilizza *system calls* per interfacciarsi con i dispositivi hardware di acquisizione).

Il modulo si interfaccia anche con altri moduli:

1. GestioneStatistiche per il trasferimento delle informazioni nel caso di annullamento dell'operazione per bolletta illegibile.
2. Timer per l'inizializzazione della temporizzazione.

END LetturaBolletta

MODULE CassaBollette
USES

EXPORTS

fun IncameraBolletta ()

Consente l'incameramento della parte di bolletta da conservare.

Comanda il passaggio (via hardware) della parte di bolletta che (dopo il taglio) dovrà essere conservata all'interno della macchina stessa. Aggiorna l'archivio dello spazio restante nella cassetta (archivio contenuto nella parte privata della classe).

fun QuantoSpazioResta ()

Fornisce la percentuale di spazio restante all'interno della cassetta per l'introduzione di nuove bollette.

Legge l'archivio posto nella parte privata.

fun SvuotaCassa ()

Consente lo svuotamento della cassa.

Questa operazione è consentita esclusivamente all'operatore. Si aggiorna l'archivio interno (sullo spazio restante nella cassetta).

END exports

IMPLEMENTATION

Questo modulo si occupa di tutte le operazioni relative alla cassa che raccoglie le bollette all'interno della macchina, al termine dell'operazione.

Come dato privato c'è l'archivio contenente lo spazio restante libero all'interno della cassetta.

END CassaBollette

MODULE GestioneBolletta
USES LetturaBolletta, CassaBollette

EXPORTS

fun PrelevaInformazioni ()

Effettua la ricognizione delle informazioni prelevabili dalla bolletta.
 Sfruttando, ovviamente, il modulo di LetturaBolletta, ritorna i valori:
 LetturaBolletta.IdentificaTipo ()
 LetturaBolletta.EstraiDati ()

fun RilasciaBolletta ()

Permette la restituzione, della fattura inserita, all'utente.
 Necessita di controllare il dispositivo di riconoscimento della bolletta, che la contiene (fisicamente) al suo interno; ciò mediante chiamata indiretta al sistema operativo (*system call*).

fun TrattaBolletta (InfoOperazione)

Abilita il trattamento della bolletta.

Utilizza i dati necessari al trattamento della bolletta (coordinate taglio, coordinate timbro, informazioni timbro), interagendo con:

1. archivio coordinate tagli bollette, presente nella parte privata del modulo stesso.
2. archivio coordinate timbro, presente nella parte privata del modulo stesso.
3. informazioni sull'operazione svolta, fornite come parametro della funzione chiamata.

Interagisce con i moduli di controllo delle periferiche:

- TagliaBolletta per il taglio della bolletta.
Parametri: *spostamento lama*
- CassaBolletta.Incamera Bolletta ()
- VidimaBolletta per la vidimazione della bolletta.
Parametri: *coordinate timbro, informazioni da scrivere*

fun QualeImporto ()

Ritorna l'importo da pagare presentato sulla bolletta.
 Si interfaccia, necessariamente, con il modulo di lettura della bolletta:
 LetturaBolletta.EstraiDati(), prelevandone solamente l'importo.

fun QualeCliente ()

Ritorna l'identificativo del cliente, presentato sulla bolletta.
 Si interfaccia, necessariamente, con il modulo di lettura della bolletta:
 LetturaBolletta.EstraiDati(), prelevandone solamente i dati sul cliente.

fun QualeTipoBolletta ()

Ritorna l'importo da pagare presentato sulla bolletta.
 Si interfaccia, necessariamente, con il modulo di lettura della bolletta:
 LetturaBolletta.IdentificaTipo ().

fun EmettiRicevuta (InfoOperazione)

Permette la stampa della ricevuta di pagamento, per l'utente.

Interagisce con la periferica di stampa mediante system call. Le informazioni trasmesse alla stampante sono ovviamente formattate in modo opportuno.

END exports

IMPLEMENTATION

E' il modulo a livello più astratto relativo al trattamento delle bollette. Si serve di alcuni sottomoduli per lo svolgimento di talune funzionalità, tali moduli sono accessibili esclusivamente da questo più generale; ogni altro modulo che necessiti informazioni attribuibili alla gestione della bolletta deve necessariamente interfacciarsi a questo.

END GestioneBolletta

MODULE Calendario

USES

EXPORTS

fun fornisciData ()

Fornisce la data attuale

Utilizza il formato: Giorno [2 cifre], Mese [2 cifre], Anno [3 cifre].

fun fornisciOrario ()

Fornisce l'orario attuale

Utilizza il formato: Ora [2 cifre], Minuti [2 cifre], Secondi [2 cifre].

END exports

IMPLEMENTATION

Consente di effettuare tutte le operazioni che coinvolgono il calendario e l'orologio. E' necessario considerare anche la possibilità di resettare l'orario o la data (in caso di errore nel mantenimento degli stessi).

END Calendario

MODULE Timer
USES Calendario

EXPORTS

fun StartTimer ()

Aziona il Timer.

Fissa la data e l'orario di inizio dell'operazione (per questo si interfaccia con il calendario: Calendario.FornisciData () e Calendario.FornisciOrario()), memorizzando questo dato in una variabile privata (il cui accesso è consentito solamente mediante le funzioni considerate in questa interfaccia).

fun StopTimer ()

Ferma il Timer.

Fissa la data e l'orario di fine dell'operazione (per questo si interfaccia con il calendario: Calendario.FornisciData () e Calendario.FornisciOrario()), memorizzando questo dato in una variabile privata (il cui accesso è consentito solamente mediante le funzioni considerate in questa interfaccia).

fun Durata ()

Calcola la durata dell'operazione (dall'orario di inizio a quello di fine, entrambi memorizzati come variabili private).

fun DataOraInizioOperazione ()

Fornisce la data e l'orario di inizio dell'operazione.

Effettua quindi la lettura del valore memorizzato come variabile ad accesso privato.

fun OraFineOperazione ()

Fornisce la data e l'orario della fine dell'operazione.

Effettua quindi la lettura del valore memorizzato come variabile ad accesso privato.

END exports**IMPLEMENTATION**

Gestisce tutte le operazioni relative alla temporizzazione dell'operazione di pagamento.

Ha diverse variabili private:

1. Data ed Ora di inizio dell'operazione
2. Data ed Ora di terminazione dell'operazione
3. Tempo massimo assegnato all'effettuazione dell'intera operazione

Quest'ultimo valore ha molta importanza essendo utilizzato per stabilire l'eventuale terminazione del tempo assegnato allo svolgimento dell'operazione di pagamento.

Il modulo, infatti, effettua un controllo sulla durata dell'operazione, quando questa supera il livello massimo stabilito si provvede automaticamente all'azzeramento del timer ed al passaggio del controllo al gestore dell'Abort:

GestorePagamento.Abort (*endTime*)

END Timer

MODULE RiconoscimentoBanconote
USES

EXPORTS

fun FornisciTipo ()

Fornisce il tipo della banconota inserita dall'utente nel lettore.

Il tipo di banconota è relativo alla programmazione dello strumento hardware di acquisizione (mediante system call), conformemente all'archivio del valore del denaro (in Gestione Denaro).

END exports

IMPLEMENTATION

Il modulo consente di riconoscere la banconota inserita nel lettore.

END RiconoscimentoBanconote

MODULE RiconoscimentoMonete
USES

EXPORTS

fun FornisciTipo ()

Fornisce il tipo della moneta inserita dall'utente nel riconoscitore di lega.

Il tipo di moneta è relativo alla programmazione dello strumento hardware di riconoscimento, conformemente all'archivio del valore del denaro (in Gestione Denaro).

END exports

IMPLEMENTATION

Il modulo consente di riconoscere la moneta inserita nel riconoscitore di lega.

END RiconoscimentoMonete

MODULE CassaDenaro
USES

EXPORTS

fun SpazioRestante ()

Fornisce la percentuale di spazio restante all'interno della cassetta.

Individua il minimo spazio per il deposito di nuovo denaro, sia banconote che monete. Utilizza il dato sullo stato di riempimento della cassetta implicitamente conservato nell'archivio del denaro in cassa (posto nella parte privata nella classe).

fun SvuotaCassa ()

Consente lo svuotamento della cassetta

Funzione utilizzabile solamente dall'operatore. Automaticamente vengono aggiornati gli archivi memorizzati nella parte privata della classe.

fun IncameraBanconota (TipoBanconota, Quantità)

Consente il deposito di banconote (dello stesso tipo) nella cassetta del denaro della macchina (le banconote provengono dal cassetto di accumulo temporaneo).

Aggiorna l'archivio sul denaro in cassa (nella parte privata della classe) in relazione ai dati passati come parametri, dà il comando al dispositivo hardware di smistamento delle banconote all'interno della cassa (sempre indirettamente mediante system call).

fun IncameraMoneta (TipoMoneta, Quantità)

Consente il deposito di monete (dello stesso tipo) nella cassetta del denaro della macchina (le monete provengono dal cassetto di accumulo temporaneo).

Aggiorna l'archivio sul denaro in cassa (nella parte privata della classe) in relazione ai dati passati come parametri, dà il comando al dispositivo hardware di smistamento delle monete all'interno della cassa (sempre indirettamente mediante system call).

fun FornisciBanconota (TipoBanconota, Quantità)

Consente la fornitura, all'esterno, delle banconote richieste

Aggiorna l'archivio sul denaro in cassa (nella parte privata della classe) in relazione ai dati passati come parametri, dà il comando al dispositivo hardware di smistamento delle banconote verso l'esterno della cassa (sempre indirettamente mediante system call).

fun FornisciMoneta (TipoMoneta, Quantità)

Consente la fornitura, all'esterno, delle monete richieste

Aggiorna l'archivio sul denaro in cassa (nella parte privata della classe) in relazione ai dati passati come parametri, dà il comando al dispositivo hardware di smistamento delle monete verso l'esterno della cassa (sempre indirettamente mediante system call).

fun QuanteBanconote (TipoBanconota)

Fornisce il numero delle banconote (del tipo passato come parametro) presenti nella cassetta.

Deve quindi, necessariamente, leggere l'archivio sul denaro in cassa (nella parte privata della classe) in relazione al tipo passato come parametro.

fun QuanteMonete (TipoMoneta)

Fornisce il numero delle monete (del tipo passato come parametro) presenti nella cassetta.

Deve quindi, necessariamente, leggere l'archivio sul denaro in cassa (nella parte privata della classe) in relazione al tipo passato come parametro.

fun QuantoDenaroTOT ()

Fornisce lo stato complessivo del denaro presente nella cassetta.

Effettua, in pratica, una copia dell'archivio sul denaro in cassa (descritto, come altri archivi utilizzati nel progetto, nel paragrafo successivo).

END exports

IMPLEMENTATION

Il modulo gestisce tutte le varie operazioni concernenti la cassetta di deposito del denaro all'interno della macchina progettata.

Nella parte privata della classe è presente l'archivio del denaro in cassa (descritto nel paragrafo successivo), il cui accesso è consentito esclusivamente mediante l'uso delle funzioni pubbliche descritte.

END CassaDenaro

MODULE CassaTMPDenaro
USES

EXPORTS

fun QuantoInTMP ()

Fornisce il valore (nell'unità di riferimento) del denaro inserito nella macchina nell'operazione in atto.

Legge, quindi, il dato (privato) contenente il valore richiesto.

fun IncrementaDenaroInserito (ValoreDenaroInserito)

Aggiorna il contatore del valore del denaro inserito dall'utente (che si accumula, appunto, nella cassetta di accumulo temporaneo), sommandovi il valore passato come parametro.

fun SvuotaTMP ()

Consente di effettuare l'operazione di svuotamento della cassetta di accumulo temporaneo del denaro (funzione accessibile solamente all'operatore addetto).

Aggiorna (azzerandolo) il contatore del denaro presente e comanda (mediante system call) l'apertura dello sportellino e la produzione del denaro all'esterno (raccoglitore).

END exports

IMPLEMENTATION

Modulo che gestisce la cassetta di accumulo temporaneo del denaro dell'utente. Effettua il conteggio del denaro inserito, consentendo di stabilire l'eventuale superamento dell'importo da pagare (il modulo di gestione del pagamento, infatti, vi si interfaccia).

La parte privata della classe include la variabile contenente il valore del denaro fino al momento introdotto.

END CassaTMPDenaro

MODULE AccettaDenaro
USES CassaTMPDenaro, RiconoscimentoBanconote, RiconoscimentoMonete

EXPORTS

fun LeggiDenaroInserito ()

Restituisce il tipo del denaro inserito.

Si interfaccia, evidentemente, con i moduli di gestione dei dispositivi di riconoscimento della banconota o della moneta inserita: il tipo è in relazione alla programmazione dei dispositivi stessi e comunque non in contraddizione con l'archivio del valore del denaro (nel modulo di gestione del denaro).

fun IncrementaDenaroInserito (Valore)

Aggiorna il contatore del denaro inserito (e depositato momentaneamente nella cassetta di deposito temporaneo) sommandovi il valore passato come parametro. Per far questo interagisce con il modulo di gestione della cassetta predetta: CassaTMPDenaro.IncrementaDenaroInserito (Valore)

fun QuantoInserito ()

Riferisce la somma del denaro inserito dall'utente (conservato nella cassetta temporanea), mediante interazione con il modulo di gestione della cassetta stessa: CassaTMPDenaro.QuantoInTMP ().

END exports

IMPLEMENTATION

Modulo che gestisce le operazioni inerenti l'aquisizione del denaro.

Si interfaccia con i moduli di riconoscimento del denaro, sia banconote (classe RiconoscimentoBanconote) che monete (classe RiconoscimentoMonete), e con il modulo relativo alla cassetta di accumulo temporaneo del denaro inserito dall'utente (CassaTMPDenaro).

END AccettaDenaro

MODULE RestituisciDenaro
USES CassaDenaro, CassaTMPDenaro

EXPORTS

fun FornisciResto (lista di TipoDenaro, Quantità)

Abilita la cassa del denaro a fornire all'esterno (all'utente) il denaro dovuto come resto per l'operazione di pagamento effettuata.

Si interfaccia quindi con la cassa del denaro mediante le funzioni:

CassaDenaro.FornisciBanconota (TipoBanconota,Quantità),

CassaDenaro.FornisciMoneta (TipoMoneta,Quantità).

fun RilasciaDenaro ()

Abilita la cassetta temporanea a rilasciare completamente il contenuto (denaro inserito dall'utente) per la terminazione anomala dell'operazione.

Naturalmente si interfaccia con il modulo di gestione della cassetta di accumulo temporaneo, mediante la funzione:

CassaTMPDenaro.Svuota().

END exports**IMPLEMENTATION**

Modulo che si occupa del sottoproblema (inquadrato all'interno del modulo di gestione del pagamento) dell'uscita del denaro dalla macchina verso l'utente.

Ovviamente si interfaccia con i moduli relativi al funzionamento delle casse del denaro (quello dell'accumulo temporaneo e quella del deposito stabile all'interno della macchina).

END RestituisciDenaro

MODULE GestioneResto
USES RestituisciDenaro, GestionePagamento, CassaDenaro

EXPORTS

fun CalcoloEmissibilità (ValoreRestoDaFornire)

Funzione che effettua il test sull'emissibilità del resto dovuto all'utente.

Test che si suddivide in alcuni passi:

1. Controllo denaro in macchina.

Portato a termine mediante l'utilizzo di alcune informazioni ricavate dall'interfacciamento con altri moduli, in particolare:

CassaDenaro.QuantoDenaroTOT (), che fornisce la quantità di denaro presente nella cassa della macchina

GestionePagamento.ConvertiDenaro (TipoDenaro), che fornisce la funzione di individuazione del valore del denaro.

2. Eventuale suggerimento di introduzione nuovo denaro all'utente.

Denaro sufficiente (in funzione del risultato del controllo precedente) a rendere possibile la fornitura all'utente dell'intero resto dovuto.

Ovviamente è necessario interagire con il modulo di gestione dell'interfaccia.

3. In base ai passi precedenti, la funzione ritorna il massimo resto fornibile all'utente (valore che si avvicina il più possibile, per difetto, al resto dovuto).

fun QuantoRestoFornito ()

Riporta in uscita il valore del resto fornito effettivamente all'utente.

fun FornisciResto (ValoreRestoFornibile)

Comando che consente la reale emissione del resto (nella quantità inserita come parametro).

Innanzitutto si scelgono quali (e quante) banconote e monete saranno fornite (di valore complessivo pari a quello da fornire, ovviamente), questo viene fatto consultando nuovamente:

CassaDenaro.QuantoDenaroTOT (), che fornisce la quantità di denaro presente nella cassa della macchina

GestionePagamento.ConvertiDenaro (TipoDenaro), che fornisce la funzione di individuazione del valore del denaro.

Successivamente si comanda alla cassa del denaro lo smistamento (in uscita) delle banconote e delle monete designate, mediante la chiamata:

RestituisciDenaro.FornisciResto (*lista di tipoDenaro, Quantità*)

fun CreditoResiduo ()

END exports

IMPLEMENTATION

Complesso modulo che coordina le diverse operazioni da effettuare per la gestione della fornitura del resto dell'operazione.

Vi sono diversi casi da contemplare ed esaminare:

- a) Il resto dovuto è fornibile completamente (rilevato dalla funzione di controllo della emettibilità): in tal caso si provvede semplicemente all'emissione dell'intera somma.
- b) Il resto dovuto non è fornibile interamente (la funzione di controllo ritorna il valore fornibile che lo approssima per difetto ed eventualmente suggerisce all'utente l'inserimento di altro denaro)
 - 1) L'utente inserisce il denaro richiesto: la macchina fornisce tutto il resto (aggiornato)
 - 2) L'utente non inserisce il denaro richiesto:
 - Conferma ugualmente l'operazione: Viene fornito il resto previsto e inserito il credito dell'utente (che può riscuotere) nella ricevuta.
 - Annulla l'operazione: L'operazione di pagamento termina in modo anomalo, viene restituita all'utente la bolletta integra ed i soldi inseriti.

END

GestioneResto

MODULE GestionePagamento
USES GestioneResto, AccettaDenaro, GestioneBolletta, GestioneStatistiche

EXPORTS

fun NuovoDenaroUtente (ValoreDiRiferimento)

E' stata inserita una nuova banconota (o moneta).
 Aggiorna il contatore del denaro mediante la chiamata al modulo di gestione dell'accettazione del denaro:
 AccettaDenaro.IncrementaDenaroInserito (*ValoreDenaro*)

fun Abort (*motivo*)

Funzione che gestisce la situazione di interruzione anomala dell'operazione.
 Si interfaccia con il modulo Timer (azzerandolo) e richiama il modulo di gestione delle statistiche per memorizzare l'operazione fallita:
 Timer.StopTime ()
 GestioneStatistiche.NuovaOperazioneFallita (*motivo*)

fun QuantoDenaroDaInserire ()

Ritorna il valore (reale) del denaro da inserire nella macchina per il pagamento.
 Ovviamente tale valore viene dato dalla somma tra il costo della bolletta da pagare (interfacendosi con il modulo di gestione delle statistiche) ed il costo dell'operazione (valore conservato nella parte privata della classe): GestioneBolletta.QualeImporto ()

fun QuantoDenaroInserito ()

Ritorna il valore (reale) del denaro inserito nella macchina dall'utente.
 Ovviamente tale valore viene dato dal denaro conservato nella cassetta di accumulo temporaneo del denaro (durante l'operazione), ci si interfaccia quindi con il modulo di accettazione del denaro:
 AccettaDenaro.QuantoInserito ()

fun QuantoRestoDaFornire ()

Ritorna il valore (reale) del resto da fornire all'utente.
 Valore richiesto al modulo di gestione del resto:
 GestioneResto.RestoDaFornire ()

fun QuantoRestoFornito ()

Ritorna il valore (reale) del resto effettivamente fornire all'utente.
 Valore richiesto al modulo di gestione del resto:
 GestioneResto.RestoFornito ()

fun QuantoCredito ()

Ritorna il valore (reale) del credito residuo dell'utente.
 Valore ottenuto dalla differenza tra il resto dovuto e quello effettivamente fornito all'utente.

fun ConvertiDenaro (TipoDenaro)

Ritorna il valore reale del denaro (codificato con l'identificatore del tipo passato come parametro) *ed il valore di riferimento* (unico usato all'interno della macchina per le motivazioni già esaminate nella parte di analisi).

Necessariamente consulta la tabella del valore del denaro (conservata all'interno della zona privata della classe e, quindi, accedibile solamente mediante questa funzione).

END exports

IMPLEMENTATION

Modulo che comprende tutte le funzionalità atte a gestire il flusso di denaro all'interno della macchina. Ogni servizio relativo strettamente al pagamento delle bolle è gestito da questo modulo.

Si fa notare che l'operazione di pagamento inizia immediatamente dopo il riconoscimento della bolletta inserita (e non all'inserimento della prima banconota) e può terminare in modo positivo (pagamento effettuato) o anomalo (pagamento abortito).

Nella parte privata della classe sono presenti:

1. Archivio costo operazione
 2. Archivio Valore Denaro
- (cnf. paragrafo successivo per una descrizione puntuale)

END GestionePagamento

MODULE SpazioMemorizzazione

USES

EXPORTS

fun DecSpazio (quantità)

Decrementa (evidentemente a seguito di una nuova memorizzazione) la variabile (privata) che rappresenta la quantità di spazio disponibile.

fun IncSpazio (quantità)

Incrementa (evidentemente a seguito di un azzeramento degli archivi) la variabile (privata) che rappresenta la quantità di spazio disponibile.

fun RiportaSpazio ()

Fornisce il valore (in percentuale) dello spazio ancora disponibile per la memorizzazione delle statistiche.

END exports

IMPLEMENTATION

Si interessa del controllo sullo spazio (su disco) per la memorizzazione delle statistiche.

END SpazioMemorizzazione

MODULE CounterOperazione
USES

EXPORTS

fun FornisciNumeroOperazione ()

Riporta in uscita il numero progressivo di identificazione dell'operazione svolta, aggiornando il contatore..

Legge, in pratica, il contenuto della variabile privata e lo incrementa di una unità.

fun Reset ()

Azzera il numero dell'operazione.

END exports

IMPLEMENTATION

Gestisce il conteggio delle operazioni svolte con successo. Le due funzioni pubbliche accedono alla variabile privata (contatore).

END CounterOperazione

MODULE StatisticheStatoOperatività
USES

EXPORTS

fun MemorizzaStato (dati)

Aggiorna l'archivio contenente le informazioni sullo stato di funzionamento della macchina.

Necessita (oltre che dello stato attuale) della data e dell'ora del cambiamento dello stato (entrambi passati nel parametro).

fun PrelevaArchivio ()

Consente di prelevare l'intero archivio (effettua quindi una copia di quello custodito nella parte privata della classe), eventualmente (se via rete) effettua controlli per l'autorizzazione.

END exports

IMPLEMENTATION

Nella parte privata della classe è inserito l'archivio sull'operatività della macchina.

Il modulo permette l'accesso controllato a questa risorsa.

END StatisticheStatoOperatività

MODULE StatisticheOperazioniFallite**USES****EXPORTS****fun** MemorizzaCaso (DatiStatistica)*Aggiorna l'archivio contenente le statistiche sui casi di operazioni non completate.**Tutte le informazioni da memorizzare (in una nuova entry dell'archivio) sono tutte passate come parametro della funzione.***fun** Statistiche (baseStatistica, tipoStatistica)*Effettua il calcolo della statistica richiesta (sulla base temporale fornita come parametro). Necessariamente utilizza l'archivio delle operazioni fallite (parte privata della classe).***fun** PrelevaArchivio ()*Consente di prelevare l'intero archivio (effettua quindi una copia di quello custodito nella parte privata della classe), eventualmente (se via rete) effettua controlli per l'autorizzazione. Utile se si preferisce effettuare all'esterno il calcolo delle statistiche.***END exports****IMPLEMENTATION***Nella parte privata della classe è inserito l'archivio delle operazioni fallite.**Il modulo permette, in pratica, l'accesso controllato a questa risorsa ed il calcolo (eventuale) di statistiche, sull'archivio stesso.***END** StatisticheOperazioniFallite**MODULE** StatisticheOperazioniEffettuate**USES****EXPORTS****fun** MemorizzaOperazione (DatiStatistica)*Aggiorna l'archivio contenente le statistiche sui casi di operazioni effettuate con successo.**Tutte le informazioni da memorizzare (in una nuova entry dell'archivio) sono tutte passate come parametro della funzione.***fun** Statistiche (baseStatistica, tipoStatistica)*Effettua il calcolo della statistica richiesta (sulla base temporale fornita come parametro). Necessariamente utilizza l'archivio delle operazioni effettuate (parte privata della classe).***fun** PrelevaArchivio ()*Consente di prelevare l'intero archivio (effettua quindi una copia di quello custodito nella parte privata della classe), eventualmente (se via rete) effettua controlli per l'autorizzazione. Utile se si preferisce effettuare all'esterno il calcolo delle statistiche.***END exports****IMPLEMENTATION***Nella parte privata della classe è inserito l'archivio delle operazioni effettuate.*

Il modulo permette, in pratica, l'accesso controllato a questa risorsa ed il calcolo (eventuale) di statistiche, sull'archivio stesso.

END StatisticheOperazioniEffettuate

MODULE Statistiche
USES Calendario, GestioneBollette, GestioneDenaro, Timer, SpazioMemorizzazione, CounterOperazione

EXPORTS
fun NuovaOperazioneEseguita ()

E' stata eseguita una nuova operazione, memorizzarne i dati nell'archivio.

Tale funzione si suddivide in alcuni passi:

1. Individua dati necessari alla memorizzazione.
Questo è possibile utilizzando funzioni pubbliche di altri moduli:
 - Timer.Durata ()
 - Calendario.DataOraInizioOperazione ()
 - GestioneBollette.QualeCliente ()
 - GestioneBollette.QualeTipoBolletta ()
 - GestioneBollette.QualeImporto ()
 - GestioneDenaro.QuantoDenaroDaInserire ()
 - GestioneDenaro.QuantoRestoFornito ()
 - GestioneDenaro.QuantoCredito ()
 - CounterOperazione.LeggiNumeroOperazione ()
2. Memorizza nuova riga nell'archivio:
StatisticaOperazioneEffettuata.MemorizzaOperazione (*dati*)
3. Decrementa spazio su disco
SpazioMemorizzazione.DecSpazio (*quantità*)

fun NuovaOperazioneFallita (*motivo*)

E' stata interrotta un'operazione, memorizzarne i dati nell'archivio.

Tale funzione si suddivide in alcuni passi:

1. Individua dati necessari alla memorizzazione.
Questo è possibile utilizzando funzioni pubbliche di altri moduli:
 - Timer.Durata ()
 - Calendario.DataOraInizioOperazione ().
2. Memorizza nuova riga nell'archivio:
StatisticaOperazioneFallita.MemorizzaCaso (*dati*)
3. Decrementa spazio su disco
SpazioMemorizzazione.DecSpazio (*quantità*)

fun NuovoStatoMacchina (*stato*)

E' mutato lo stato di operatività della macchina, memorizzarne i dati nell'archivio.

Tale funzione si suddivide in alcuni passi:

1. Individua dati necessari alla memorizzazione.
Questo è possibile utilizzando funzioni pubbliche di altri moduli:
 - Calendario.DataOraFineOperazione ().
2. Memorizza nuova riga nell'archivio:

3. StatisticaStatoOperatività.MemorizzaStato (*dati*)
 Decrementa spazio su disco
 SpazioMemorizzazione.DecSpazio (*quantità*)

fun RichiestaStatistiche (*baseStatistica, tipoStatistica*)

Effettua il calcolo della statistica richiesta (sulla base temporale fornita come parametro).

A seconda del tipo di statistica richiesta richiama le funzioni:

- StatisticaOperazioniEffettuate.Statistiche (*baseStatistica, tipoStatistica*),
- StatisticaOperazioniFallite.Statistiche (*baseStatistica, tipoStatistica*)

fun PrelevaArchivio (*tipo*)

Consente di prelevare l'intero archivio (effettua quindi una copia di quello custodito nella parte privata della classe), eventualmente (se via rete) effettua controlli per l'autorizzazione. Utile se si preferisce effettuare all'esterno il calcolo delle statistiche.

A seconda del tipo (parametro) si attiva una delle funzioni:

- StatisticaOperazioniEffettuate.PrelevaArchivio ()
- StatisticaOperazioniFallite.PrelevaArchivio ()
- StatisticaStatoOperatività.PrelevaArchivio ()

END exports

IMPLEMENTATION

Modulo che gestisce le molteplici operazioni effettuate sulle statistiche (memorizzazione, estrazione,...). Tutti i moduli che hanno necessità di interagire con funzionalità relative alla gestione complessiva delle statistiche devono interfacciarsi con questo modulo.

Questa classe non ha strutture dati nella parte privata, è da osservare però che tutti gli archivi delle varie statistiche sono comunque nelle parti private dei sottomoduli che a questa fanno riferimento.

END Statistiche

MODULE Monitor
USES CassaDenaro, CassaBollette, SpazioMemorizzazione, Statistiche
EXPORTS
fun Recognize ()

1. *Funzione che effettua la ricognizione completa sull'intera macchina*, testando le varie periferiche, controllando i valori riportati dai sensori ed effettuando controlli anche sugli archivi relativi a particolari dispositivi.
 - Il testing dei dispositivi viene effettuato mediante interrogazioni indirette tramite system calls. I sensori interpellati sono relativi a:
 - a) Stato presenza inchiostro nella stampante
 - b) Stato presenza carta per stampa ricevute
 - c) Stato alimentazione (elettrica) della macchina
 - d) Stato generale funzionamento periferiche (stampante, lettore bollette, lettore banconote, riconoscitore di lega,...)
 - Il controllo sulle informazioni riportate negli archivi tiene invece conto di:
 - e) Spazio su disco
mediate chiamata: SpazioMemorizzazione.LeggiSpazio ()
 - f) Spazio nella cassetta delle bollette
mediate chiamata: CassaBollette.QuantoSpazioResta ()
 - g) Spazio nelle cassette del denaro
mediate chiamata: CassaDenaro.SpazioRestante ()
2. Aggiorna lo stato attuale del sistema (stato memorizzato nella parte privata della classe stessa), questo aggiornamento (se comporta modifica rispetto lo stato precedente) sarà riportato nelle statistiche mediante la chiamata: Statistiche.NuovoStatoMacchina (stato)
3. Invia le informazioni sul nuovo stato (previa opportuna formattazione) al modulo di interfacciamento:
Interfacciamento.StatoOperatività (stato)

END exports

IMPLEMENTATION

Si tratta dell'unico modulo (particolarmente complesso ed articolato, ma comunque anche logicamente unico) che permette di effettuare il monitoraggio dell'intero sistema. L'unica funzione sfrutta anche una variabile (contenente lo stato attuale del sistema) memorizzata nella parte privata della classe. La funzione, chiamata dal modulo che impone l'inizio effettivo dell'operazione (modulo di gestione della bolletta), ritorna il valore dell'abilitazione all'effettuazione dell'operazione stessa (operazione abilitata o non abilitata).

END Monitor

MODULE Interfacciamento
USES GestioneBolletta, GestionePagamento

EXPORTS
fun Stato (nuovoStato)

Permette la modifica sul video dello stato della macchina (una zona del video viene infatti riservata alle informazioni sulla situazione attuale del sistema). Questa funzionalità viene richiamata direttamente dal modulo di monitoraggio della macchina.

fun Pagamento (SituazioneAttuale)

Effettua la ricognizione (sul modulo pagamento) dello stato attuale dell'operazione di pagamento, memorizza eventualmente i dati parziali su una locazione di memoria dedicata a questo scopo, e visualizza le informazioni sulla situazione del pagamento.

La connessione con il modulo di gestione del pagamento avviene mediante:

GestionePagamento.DenaroInserito ()

GestionePagamento.RestoDaFornire ()

GestionePagamento.RestoFornito ()

GestionePagamento.CreditoResiduo ()

fun DatiBolletta (dati)

Effettua la lettura (sul modulo di gestione della bolletta) di alcuni dati della bolletta, memorizza eventualmente i dati parziali su una locazione di memoria dedicata a questo scopo, e visualizza le informazioni sulla bolletta.

La connessione con il modulo di gestione della bolletta avviene mediante:

GestioneBolletta.QualeImporto ()

GestioneBolletta.QualeTipoDiBolletta ()

fun OnAudio ()

Funzione che abilita la produzione del messaggio audio (in pratica la lettura della guida alle operazioni da effettuare).

Aggiorna la variabile interna (privata) che tiene conto, appunto, dello stato attuale dell'abilitazione all'emissione dell'audio.

END exports

IMPLEMENTATION

Importantissimo modulo che si occupa della produzione dell'interfacciamento verso l'utente: interfacciamento video (mediante schermate testuali, grafici ed animazioni) ed interfacciamento audio (eventualmente richiesto dall'utente).

Stabilire se al momento è richiesto una produzione di messaggi audio consiste semplicemente nel testare la variabile interna (privata) dedicata allo stato audio.

Tutte le funzioni fanno riferimento a degli archivi la qualità delle cui informazioni è particolarmente importante per l'intero progetto.

Questi archivi (esaminati nel paragrafo successivo) sono privati della classe, accessibili quindi esclusivamente mediante le funzioni sopra riportate, e sono:

1. archivio stringhe messaggi
2. archivio sequenze video
3. archivio messaggi audio

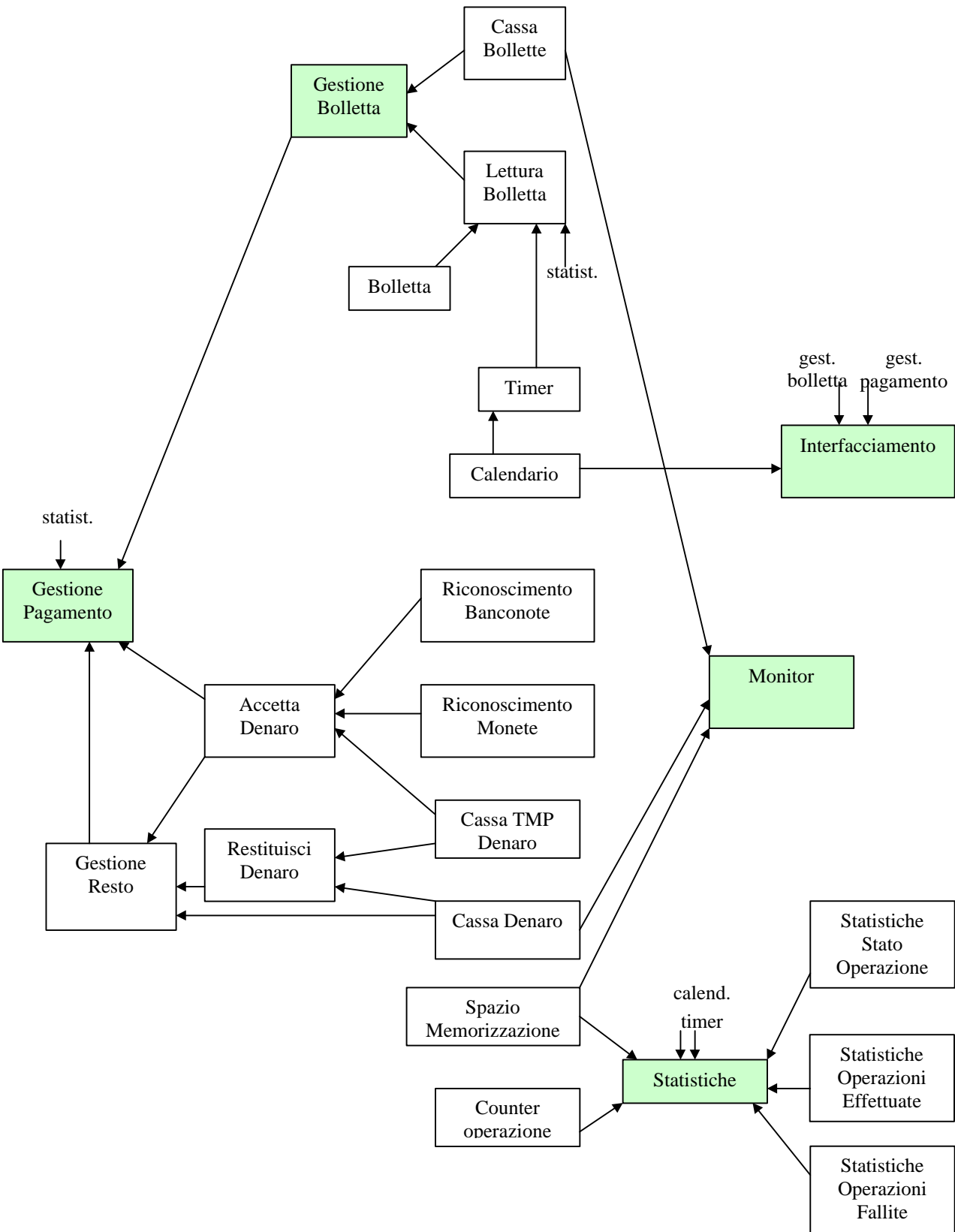
L'individuazione delle informazioni corrispondenti al messaggio che si vuole dare all'utente, è realizzata mediante l'uso di un codice messaggio.

Il modulo permette la visualizzazione della data e dell'orario, si interfaccia automaticamente quindi con il calendario:

Calendario.FornisciData ()

Calendario.FornisciOra ()

END Interfacciamento



Archivi utilizzati

Analisi degli archivi utilizzati

Di seguito sono elencati gli archivi che vengono utilizzati all'interno del progetto a p r i l. Si effettuerà una breve descrizione della loro funzione e della loro struttura interna.

1. Archivio maschere fatture

Si tratta di un archivio utilizzato nel modulo di gestione delle bollette (per il riconoscimento del tipo). Consiste in una serie di immagini di maschere di fatture, con le quali viene confrontata quella della bolletta inserita, individuandone il tipo corrispondente.

2. Archivio codici a barre

E' utilizzato per la decodifica dei codici a barre estratti dall'immagine della fattura (nel modulo di gestione delle bollette) e consiste in una lista di records a due campi: tipo codice a barre (che lo identifica) e tipo decodifica (che rappresenta il codice del modo con il quale il codice a barre deve essere decodificato).

3. Tempo massimo operazione di pagamento

Si tratta di un record solamente, può essere modificato esclusivamente dall'operatore, non viene quindi aggiornato con l'effettuazione delle operazioni. Contiene un numero intero indicante (in secondi) il tempo massimo entro il quale deve terminare l'operazione di pagamento.

4. Archivio costo operazione pagamento

Anche questo è in realtà un solo record. Memorizza il costo (nell'unità di riferimento considerata nella macchina) dell'effettuazione dell'operazione.

L'importo di questo costo –inserito e modificato esclusivamente dall'operatore- deve essere concordato ed accettato dalle autorità competenti (potenzialmente potrebbero non esserci costi aggiuntivi).

5. Codice ID macchina

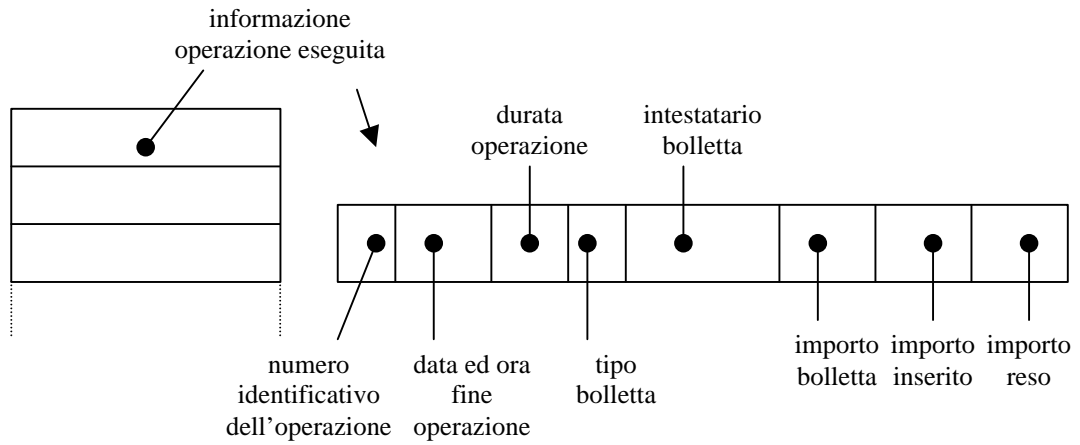
Si tratta di un record contenente una stringa che rappresenta la macchina stessa (codice identificativo). Tale record non è accessibile in scrittura.

6. Archivio valore denaro

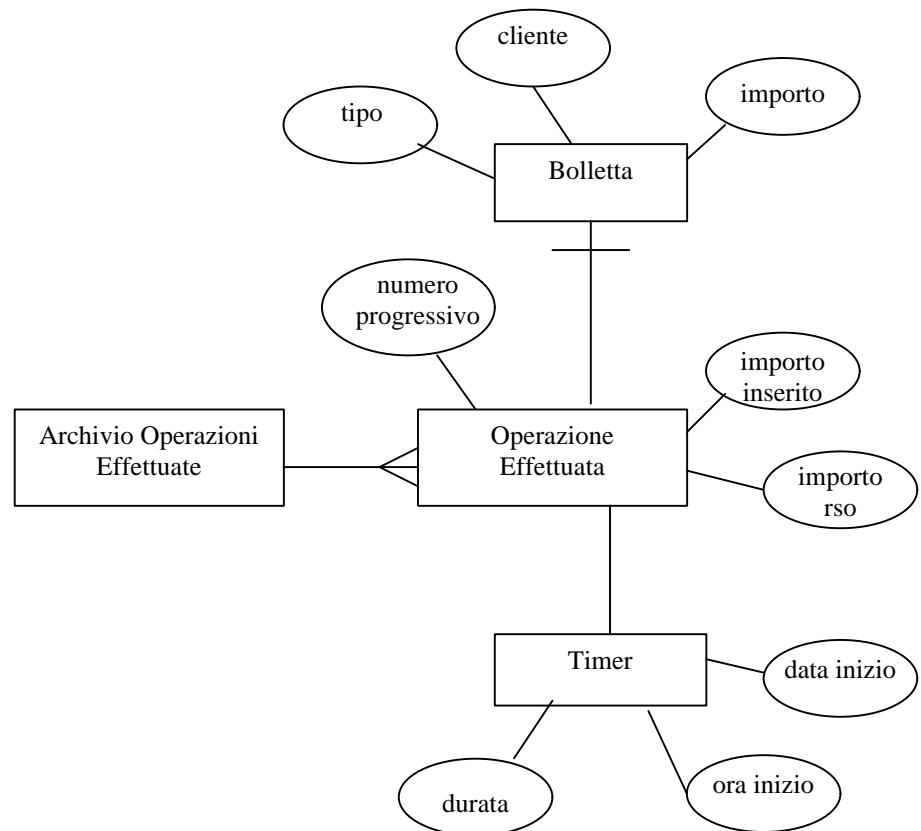
Si tratta di una tabella che permette di stabilire, dato il tipo di banconota o moneta (individuato dalla posizione nella tabella stessa), il valore nell'unità di riferimento. Per completezza si inserisce anche il valore reale nel mercato (dato comunque mai utilizzato all'interno della macchina):

	valore nell'unità di riferimento interna	valore reale
posizione 1 → (banconota tipo 1)	0.5025	Lire 1000
	1.0050	Lire 2000

Ogni riga del file contiene le informazioni in relazione ad una operazione effettuata con esito positivo (quelle relative ad operazioni fallite saranno memorizzate in un file dedicato):



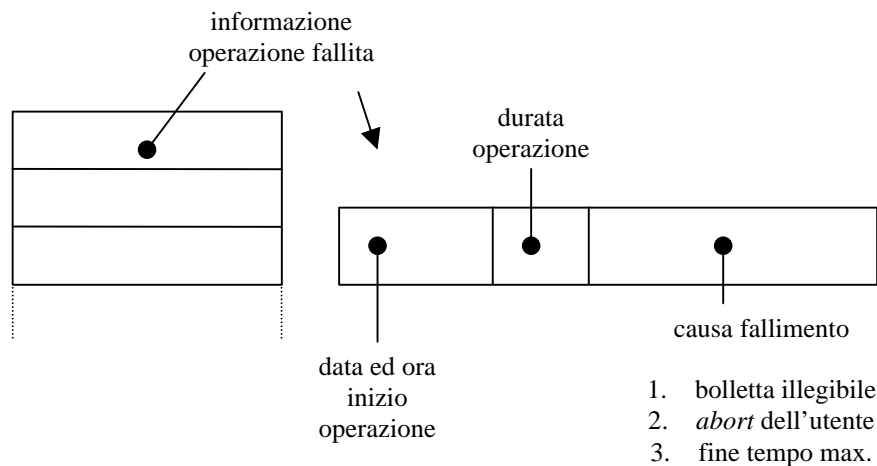
ERD



11. Archivio generale delle operazioni fallite

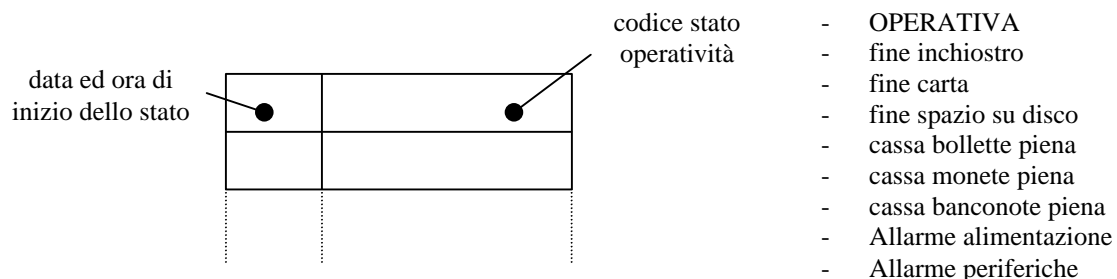
Memorizza le informazioni relative alle operazioni che non hanno avuto esito positivo. Vi si può accedere in lettura esclusivamente mediante i privilegi posseduti dall'operatore, mentre l'accesso in scrittura è consentito all'interno del modulo di gestione delle statistiche.

Consiste in una tabella nella quale ogni *entry* rappresenta un'operazione fallita:



12. Archivio operatività macchina

E' un archivio dove vengono immagazzinati gli stati di operatività della macchina. Ogni riga di questa tabella rappresenta quindi un mutamento di stato di operatività dovuto al verificarsi di qualche fatto (negativo, quale ad esempio la terminazione dello spazio di memorizzazione su disco, o positivo, quale può essere l'intervento di ripristino effettuato dal personale della manutenzione):



13. Contatore progressivo numero identificativo operazione

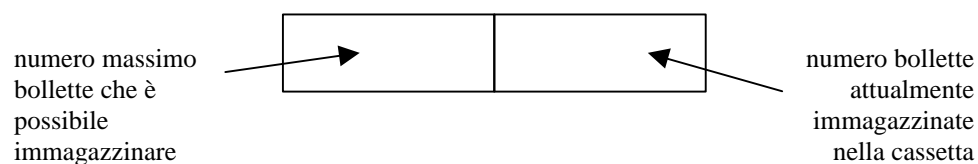
E' semplicemente un record contenente il numero di identificazione dell'operazione (viene incrementato al termine di una operazione effettuata con successo).

14. Spazio su disco

Indica il restante spazio su disco per la memorizzazione dei dati statistici. Viene decrementato ad ogni memorizzazione, si incrementa solo quando l'operatore esterno preleva tutti i dati (facendone una copia) che -quindi- possono essere cancellati.

15. Stato cassa bollette

E' un record contenente due campi:



16. Archivio stringhe messaggi

Contiene una lista di stringhe (che potranno essere visualizzate sullo schermo), identificate dalla loro posizione nella lista stessa.

17. Archivio sequenze video

Contiene una tabella nella quale si fa corrispondere ad un determinato codice di messaggio (che identifica il tipo di messaggio da visualizzare), il codice identificativo dell'immagine o della sequenza video.

18. Archivio messaggi audio

Contiene una tabella nella quale si fa corrispondere ad un determinato codice di messaggio (che identifica il tipo di messaggio da visualizzare), il codice identificativo della sequenza audio.

19. Memoria interna

Si tratta di un semplice buffer dove memorizzare temporaneamente informazioni (provenienti da diversi moduli) per un loro successivo utilizzo organico.

Indicazioni e riferimenti bibliografici

references

Ingegneria del Software

- [1] A. Fuggetta, C. Ghezzi, S. Morasca, A. Morzenti, M. Pezzè
Ingegneria del Software Settembre, 1996
Arnoldo Mondadori Editore S.p.A.
- [2] Pressman
Principi di Ingegneria del Software Febbraio, 1996
McGraw-Hill

Ricerche nel settore

- [3] Informazioni sulla struttura, sulle funzionalità, sull'utilizzo ed sui dispositivi di lettura e decodifica presenti sul mercato, dei codici a barre
www.barcode.it
- [4] Esempi di macchine con funzionalità simili a quella progettata, presenti nella città di Palermo
macchine per il pagamento di fatture Telecom
c/o *Telecom Italia S.p.A.* – Filiale di Palermo via A.Pacinotti, 57 90145 Palermo
Ditta costruttrice: **sigma**, www.sigma.com

Strumenti di sviluppo

- [5] *Upper CASE* (Computer Aided Software Engineering)
turbo CASE /Sys Ver2.02 ©StructSoft, Inc. Trial version (45 days)
www.turbocase.com
- [6] Strumento di pianificazione della progettazione
Project 98 ©Microsoft Corporation SR-1 Evaluation copy
www.microsoft.com/project
- [7] Editor grafico-testuale
Word 97 ©Microsoft Corporation
www.microsoft.com/office
- [8] Strumento per la visualizzazione, la compressione e l'impostazione alla stampa dell'intera documentazione presentata
Acrobat 3.01 ©Adobe Systems Inc.
(Acrobat Distiller 3.01, Acrobat Reader 3.0)
www.adobe.com/acrobat

Ricerche sulle componenti hardware

- [9] Dispositivi di riconoscimento di lega e di lettura banconote
Coges s.r.l.
www.coges.it E-mail: coges@goges.it

Indice

Index

I Analisi del sistema

1.	Ambiente di Sviluppo	4
2.	Analisi dei Requisiti	5
-	Obiettivi del Sistema	5
-	Vincoli temporali ed economici	5
-	Requisiti imposti dal Committente	5
3.	Studio di Fattibilità	6
-	Introduzione	6
-	Alternative possibili	6
-	Descrizione del Sistema	6
-	Analisi Costi-Benefici <i>fattibilità economica</i>	7
-	a) Benefici	7
-	b) Costi	7
-	Valutazione dei rischi tecnici <i>fattibilità tecnica</i>	8
-	Implicazioni legali <i>fattibilità legale</i>	8
4.	Analisi del Sistema	9
-	Stato attuale del sistema di pagamento	9
-	Descrizione di massima del nuovo sistema	10
-	Analisi degli Obiettivi	10
-	Impatto del nuovo metodo	10
-	a) automatico	10
-	b) semplice	11
-	c) veloce	11
-	d) senza restrizioni di tempo	11
-	Interfacciamento con l'utente	12
-	a) aspetto esterno	12
-	b) usabilità dai portatori di handicap	12
-	c) usabilità da chi ha difficoltà visive o di comprensione dei messaggi video	13
-	d) user-friendly	13
-	Tecnologie necessarie	15
-	1) per l'analisi del sistema	15
-	2) per il progetto del sistema	15
-	3) per la codifica	15
-	4) per la realizzazione finale della macchina	15
-	- dispositivo per il riconoscimento delle banconote	15
-	- dispositivo per il riconoscimento delle monete	15
-	- dispositivo per la fornitura del resto	16
-	- dispositivo per l'acquisizione delle immagini delle fatt.	17
-	- dispositivo per la decodifica dei codici a barre	17
-	- dispositivo per il taglio delle fatture	17
-	- dispositivo di visualizzazione	17
-	- dispositivo audio	17

- Sviluppi possibili	18
A. Installazione di un gestore automatico di turno	18
B. Cambio Valuta	19
C. Cambio Lingua	20
D. Immissione di nuovi tipi di banconote/monete	20
E. Cancellazione di banconote/monete ritirate dalla circolazione	20
F. Immissione di nuovi tipi di fatture gestibili dalla macchina	20
G. Cancellazione dei tipi di fatture da non dover più gestire	20
H. Connessione in rete locale	21
I. Connessione in rete globale	21

II Pianificazione

5. Risorse assegnate al progetto	24
- risorse umane	24
interne al team di sviluppo	
1) responsabile progetto	24
2) analista	24
3) ingegnere del software	24
4) responsabile codifica	25
5) programmatori	25
esterne al team di sviluppo	
6) esperto di psicologia e comportamento umano	25
7) esperto di dispositivi hardware	26
8) esperto di affari legali	26
- Risorse hardware	27
- Risorse software	28
6. Resouce Sheet	30
7. Task Usage	31
8. Resouce Usage	36
9. Gantt Chart	40
10. Pert Chart - livello più astratto	43
11. Pert Chart	44

III Analisi del software

12. Funzionalità del progetto	46
13. Data Flow Diagrams	47
- Context Diagram	47
- DFD/0 progetto april	48
- DFD/1 effettuazione operazione	49
- DFD/1.1 lettura bolletta	50

-	DFD/1.2	pagamento	54
-	DFD/1.2.4	trattamento del resto	58
-	DFD/1.3	trattamento bolletta ed emissione ricevuta	62
-	DFD/2	statistiche	66
-	DFD/2.1	memorizzazione statistiche	71
-	DFD/3	verifica operatività macchina	77
-	DFD/4	presentazione informazioni	84
14.	Requirement Dictionary		87
15.	State Transition Diagram		95
 IV Progettazione del Software			
16.	Modularizzazione del progetto		96
-	Enhanced Data Flow Diagram		97
-	Architecture Context Diagram		98
-	Architecture Flow Diagram		99
-	Architecture Interconnect Context Diagram		100
-	Architecture Interconnect Diagram		101
17.	Modularizzazione del software.....		102
-	Scelta del linguaggio di codifica		102
-	Specifiche TDN		103
-	Diagramma delle chiamate dei moduli		123
18.	Archivi utilizzati		124
-	Analisi degli archivi utilizzati		124
 Riferimenti bibliografici			129