

**NSIP  
Manutenzione MAC e MEV**

**Sistema di sviluppo**

**ISY**

**Sistema di sviluppo software multiambiente**

**INFORMAZIONI GENERALI**

Documento : ISYDGEN0  
Aggiornamento : 0  
Data : 21/01/95

## Sommario

<b>INTRODUZIONE</b> .....	1
<b>LO SVILUPPO</b> .....	4
<b>ARCHITETTURA DEL SISTEMA ON-LINE</b> .....	7
GENERALITÀ .....	7
STRUTTURA DELLE TRANSAZIONI CONVERSAZIONALI (CICS) .....	7
COLLEGAMENTO TRA PROGRAMMI .....	8
ACQUISIZIONE PARAMETRI DI ATTIVAZIONE TRANSAZIONI .....	9
VISUALIZZAZIONE LISTE E SELEZIONE DATI .....	11
TASTI FUNZIONALI .....	12
STAMPE ON-LINE .....	12
<b>ARCHITETTURA DEL SISTEMA BATCH</b> .....	13
GENERALITÀ .....	13
ACQUISIZIONE PARAMETRI DI ATTIVAZIONE.....	13
<b>LA CENTRALIZZAZIONE DEI CONTROLLI</b> .....	14
IL CONTROLLO ED EDIT DEI CAMPI IN ON-LINE .....	14
IL CONTROLLO ED EDIT DEI CAMPI IN BATCH.....	14
<b>I SERVIZI PER L' UTENTE</b> .....	16
SECURITY.....	16
HELP .....	16
MENU .....	18
DIZIONARIO DATI.....	18
RICHIESTE OFF-LINE.....	19

## Introduzione

L'utilizzo dell'informatica ha subito negli anni un'espansione estremamente rapida, sia per la crescente disponibilità dell'hardware, sia per la sempre maggiore esigenza delle aziende di disporre di strumenti atti a ridurre i costi di gestione e a fornire informazioni tempestive e corrette a supporto delle decisioni.

La richiesta di nuove applicazioni software da parte degli utenti rimane spesso disattesa o soddisfatta solo parzialmente, malgrado il notevole impegno profuso dagli specialisti del settore.

Conseguentemente gli utenti hanno spesso la sensazione che la produttività del settore EDP sia bassa. Ciò è spesso vero e per giunta il tempo dedicato alla manutenzione e agli ampliamenti successivi risulta percentualmente troppo elevato rispetto al tempo dedicato allo sviluppo del sistema.

L'instaurarsi di questa situazione è spesso imputabile ai seguenti fattori:

- le esigenze degli utenti non sono state completamente definite, valutate o comprese prima di iniziare l'impostazione del progetto;
- l'individuazione fra le varie alternative possibili non è stata fatta attraverso una approfondita analisi;
- i gruppi di lavoro non dispongono di cultura e di strumenti per la pianificazione e il controllo dell'avanzamento dei progetti;
- spesso la fuoriuscita di una risorsa dal gruppo di lavoro provoca contraccolpi di difficile gestione, essendo la figura specifica quasi sempre unico punto di riferimento per quel tipo di attività;
- la documentazione, quando viene prodotta, è realizzata a progetto ultimato secondo criteri individuali;
- la progettazione delle applicazioni viene effettuata con modalità e tecniche personali, piuttosto che con adeguate metodologie standard predefinite.

In generale si può affermare che i progettisti tendono a concentrarsi prevalentemente sugli aspetti tecnici, trascurando gli aspetti metodologici e organizzativi.

Per migliorare l'efficienza produttiva e l'efficacia degli interventi degli specialisti EDP, oltre ad applicare una corretta e rigorosa metodologia di analisi, è fondamentale l'adozione di una metodologia standard soprattutto per la fase di sviluppo dei sistemi informativi.

Normalmente lo sviluppo è dipendente dall'ambiente di esercizio del sistema che deve essere realizzato.

Il sistema ISY costituisce una metodologia che definisce regole standard e comprende una serie di strumenti che hanno l'obiettivo di migliorare la qualità del software prodotto, aumentando nello stesso tempo la produttività sia nella fase della programmazione che in quella del successivo test.

Il sistema, tramite l'impiego di particolari tecniche di generazione del software, cerca anche di svincolare, per quanto possibile, lo sviluppo dei programmi, dai particolari prodotti di sistema utilizzati sull'elaboratore impiegato, prevedendo comunque l'uso del linguaggio di programmazione COBOL.

L'adozione del presente sistema comporta il rigoroso rispetto di tutte le norme e convenzioni descritte nella documentazione dello stesso. In cambio offre una serie di vantaggi e rende disponibili alcuni strumenti preziosi per il programmatore, nella fase di sviluppo, ed altri, a corredo della applicazione realizzata, che saranno utilissimi per l'utente finale nella fase di esercizio.

I principali vantaggi, relativi allo sviluppo, derivanti dall'uso del sistema ISY si possono sintetizzare nei seguenti punti:

- rigorosa omogeneità di tutto il software applicativo realizzato;
- riduzione dei tempi di programmazione e semplificazione del processo di sviluppo, tramite l'utilizzo di modelli di riferimento e l'adozione di componenti standard dell'architettura del sistema;
- riduzione dei tempi per il test, che sarà limitato esclusivamente alla parte logica del programma;
- facile portabilità del software applicativo, tra vari Sistemi Operativi;
- disponibilità di un dizionario dei dati dell'applicazione, acceduto simultaneamente dalle routines di controllo degli stessi e dal sistema di documentazione on-line;
- facile manutenzione del software sviluppato;
- programmi di stampa operanti simultaneamente in ambiente batch e on-line.

Gli strumenti più importanti utilizzabili dal programmatore consistono in:

- numerose routines di sistema di uso comune;
- componenti standard (come definizioni di aree comuni o parti di programma standard) già disponibili sotto forma di COPY;
- ausili automatici per la generazione del software, i quali consentono di collegare alle istruzioni specifiche della logica da sviluppare, schemi standard di riferimento predefiniti, dipendenti dal tipo di programma e dall'ambiente target di esercizio.

Gli strumenti utilizzabili dall'utente finale, accessori all'applicazione realizzata, sono principalmente i seguenti:

- sistema di security basato sull'abilitazione selettiva sia alle transazioni on-line che alla lettura/scrittura di classi di dati;
- menù gerarchico delle transazione cui l'utente è abilitato;
- help on line delle transazioni comprendente help selettivo sui campi e la possibilità di trasferire l'input dai valori possibili elencati;
- sistema di richiesta on-line di elaborazioni batch con l'introduzione dei parametri di elaborazione;

- › possibilità di stampa on-line di tutti gli elenchi visualizzati.

I documenti che illustrano il Sistema di sviluppo multiambiente sono:

- **Informazioni generali (ISYDGEN)** costituito dal presente documento che contiene considerazioni generali ed introduttive sui principali aspetti del sistema di sviluppo. In particolare viene spiegato come si riesce ad ottenere la possibilità di far operare il software in ambienti operativi diversi. Quindi viene illustrata l'architettura del sistema in ambiente on-line (attualmente CICS) e in ambiente batch, viene sottolineata l'importanza ed il significato della centralizzazione del controllo e conversione dei dati, viene accennato alle modalità di generazione automatica dei programmi, vengono elencati i principali servizi aggiuntivi ottenuti con l'impiego di tale sistema.
- **Manuale per il programmatore (ISYDPRG)** illustra le modalità di preparazione delle *Specifiche*. In particolare vengono esaminate dettagliatamente la parte delle Specifiche di base e la parte delle Specifiche di routine, descritte alcune tecniche particolari di programmazione, illustrati sinteticamente gli *Schemi* disponibili, mostrate le procedure operative di generazione di sorgenti, eseguibili, documentazione nei diversi ambienti.
- **Manuale di sistema (ISYDSYS)** contiene la documentazione tecnica per la manutenzione del software di sistema e illustra il linguaggio per la scrittura di nuovi Schemi. Tale documento quindi entra più in dettaglio nella descrizione del funzionamento delle varie parti del sistema, illustrando le utilities impiegate e le regole applicate (più o meno implicitamente) da tutti i programmi che vengono sviluppati in tale ambiente.
- **Manuale per l'utente (ISYDUTE)** illustra i servizi forniti agli utenti finali, con i sistemi informativi sviluppati in tale ambiente. In particolare riporta la documentazione delle transazioni e delle elaborazioni batch disponibili presentando anche esempi di pannelli video e di tabulati ottenibili.
- **Norme di codifica (ISYDCOD)** contiene le regole di denominazione di tutti i componenti dei sistemi informativi sviluppati nell'ambiente standard di cui si parla.
- **Descrizione dei dati (ISYDDAT)** descrive in dettaglio i dati di sistema, siano files, record, campi, oppure COPY Cobol impiegati nei vari moduli del sistema.
- **Architettura del software di sistema (ISYDARC)** è l'output prodotto dal sistema di documentazione IDS che mostra la struttura gerarchica di tutti i componenti (dai documenti alle routines) del sistema di sviluppo.

E' opportuno inoltre tenere presenti i seguenti altri documenti:

- **Specifiche di struttura dei programmi in ambiente batch (ISYDBAT)** fornisce utili indicazioni per strutturare in maniera elegante ed omogenea programmi di elaborazione batch.
- **Sistema di documentazione software (IDS MAN)** è il documento che descrive il sistema per documentare la configurazione strutturale delle applicazioni sviluppate.

## Lo sviluppo

Il sistema di sviluppo in oggetto è qualificato come *multiambiente* perchè le applicazioni, tramite esso realizzate, hanno la possibilità di essere facilmente *portate* in ambienti operativi diversi da quello nel quale sono state originariamente sviluppate. Questa portabilità è ottenibile grazie a due fatti principali:

- nei programmi da sviluppare sono state separate le parti dipendenti dal sistema operativo (in pratica dal monitor on-line) dalle parti applicative;
- le parti dipendenti dal sistema operativo sono state rese standard e predisposte (o predisponibili) per i diversi ambienti operativi nei quali si vuole operare.

Gli ambienti operativi utilizzabili, ad oggi, sono tutti quelli nei quali esiste una versione del CICS e (prossimamente, maggio '94) l'ambiente VM dell'IBM, e l'ambiente UNIX nativo. Si fa riferimento principalmente a sistemi interattivi, prima di tutto per la loro maggiore complessità, e poi perchè la realizzazione di applicazioni moderne dovrà essere prevalentemente effettuata secondo questa modalità.

I primi coprono, non solo praticamente tutte le macchine IBM e i relativi possibili sistemi operativi (VSE, MVS, OS/2, ecc.) ma anche tutti gli ambienti UNIX per il quale è disponibile un emulatore Microfocus del CICS.

Si vuole evidenziare che la portabilità delle applicazioni è tanto più possibile quanto più è standard e portabile il DBMS utilizzato. Tutte le operazioni di I/O infatti stanno nelle parti applicative dei programmi, quindi usare ad esempio files VSAM potrebbe creare più problemi, da questo punto di vista, che usare un DBMS basato sull'uso del SQL standard (SQL/DS, DB2, ORACLE, ecc.).

Per comprendere questa possibilità di separare le parti applicative dei programmi dalle parti di sistema, si può dire che le architetture dei programmi possono essere classificate in un numero abbastanza ristretto di categorie. Ognuna di queste categorie richiederà la presenza di determinate funzionalità all'interno dei programmi, cioè di particolari modi di agire a fronte di condizioni fornite dall'utente o dai dati. Questo vuol dire che in ognuna di queste tipologie di programmi dovranno esistere dei blocchi di istruzioni pressochè uguali per realizzare analoghe funzioni. D'altro canto ogni categoria di programma presenterà anche delle modalità operative identiche.

Per esemplificare, un programma che realizza una funzione di acquisizione dati sarà sempre basato su un ciclo che prevede l'invio del pannello, la ricezione dei dati digitati, il loro controllo con l'emissione dei messaggi di errore e la ripetizione del ciclo in caso di errore o l'esecuzione dell'acquisizione in caso di dati validi.

La differenza fra l'aspetto del programma ed un altro appartenente alla stessa categoria sarà quindi limitata al pannello di acquisizione, all'I/O, a controlli particolari sui dati, a funzioni aggiuntive particolari.

La separazione logica e fisica delle parti applicative dei programmi da quelle di sistema, ha prodotto anche l'effetto di standardizzare e omogeneizzare al massimo la produzione dei programmi stessi. Di fatto queste parti di sistema, denominate *Schemi scheletro* contengono non solo l'interfaccia col monitor on-line (per esempio tutte le Send e Receive dei pannelli), ma anche la parte architeturale del programma stesso, cioè, in

sostanza, la tipologia base di elaborazione del programma (per esempio: visualizzazione di una lista, maintenance con un pannello, ecc.). Tali tipologie, come detto, sono sempre in numero assai limitato, e anzi la loro diversità spesso purtroppo dipende dalla differente capacità dei programmatori, per cui per scrivere due programmi sostanzialmente simili spesso le tecniche di programmazione sono completamente diverse.

Di questi Schemi scheletro, quindi, per ogni ambiente che lo richiede, ne sono stati preparati un tantum un certo numero. Questo, unito al fatto di aver creato tutta una serie di utilities richiamate dagli schemi stessi, ha permesso di concentrare la massima attenzione sul loro sviluppo, inserendo al loro interno svariate funzionalità anche di un certo grado di complessità.

Ci sono quindi tanti schemi scheletro per quante sono le categorie di programmi necessarie e ciascuno conterrà in forma standard, completa, ordinata ed efficiente tutte le istruzioni che altrimenti dovrebbero essere riscritte, e spesso anche ritestate, in ogni programma applicativo.

Le Specifiche che dovranno essere fornite per la realizzazione dei programmi possono essere quindi limitate alla definizione della categoria di appartenenza del programma stesso e di tutti gli aspetti particolari che sono richiesti.

Questa separazione fisica delle parti dei programmi è stata resa possibile dal contemporaneo sviluppo di un sistema per la loro successiva fusione per la generazione del sorgente Cobol da compilare. Tale sistema è costituito essenzialmente da:

- programma Generatore che legge in input le *Specifiche* (parti applicative) e gli *Schemi* (parti di sistema) e le fonde per creare il sorgente Cobol;
- linguaggio di controllo che guida l'attività del Generatore, inserito tra le istruzioni Cobol degli schemi
- norme di programmazione che regolano lo sviluppo applicativo delle Specifiche

Con questa tecnica si possono ottenere anche 'risparmi' nella produzione del software. Si consideri, ad esempio, che da una Specifica per un programma di stampa si possono ottenere sia la versione batch che la versione on-line dello stesso programma, oppure da uno schema di visualizzazione a lista on-line si può ottenere un programma che stampa la stessa lista in batch.

Questo sistema può offrire anche altre funzionalità, oltre alla generazione dei programmi, scrivendo di volta in volta schemi specifici. Ad esempio può essere prodotta la documentazione tecnica dei programmi a partire dalle *Specifiche*, oppure può essere generata la job stream per il caricamento della documentazione on-line delle transazioni, oppure possono essere realizzate delle cross-reference dei campi di date base utilizzati nei programmi di un progetto e così via.

A questo si aggiunga che il Generatore, essendo un programma in Cobol che legge e scrive files sequenziali, può operare in differenti ambienti compreso il Personal Computer. Le modalità di funzionamento del programma generatore sono facilmente desumibili dalla descrizione, riportata nel Manuale di sistema, del linguaggio utilizzato nello schema scheletro.

L'utilizzo di suddetti strumenti, linguaggi, norme comporta il rigoroso rispetto di precisi standard di sviluppo per le parti di competenza dei programmatori, ma nello stesso tempo implica tale rispetto per le parti di sistema già sviluppate.

Queste parti sono costituite, oltre che dagli Schemi già citati, anche da un numero considerevole di utilities che sono chiamate a svolgere i compiti più ripetitivi nei vari punti dell'architettura del sistema.

Ruolo fondamentale assume, a tale proposito, la controllo e la conversione dei dati centralizzata, che permette di concentrare in un unico programma, identico in ambiente batch e in ambiente on-line, tutti i controlli formali dei dati (in input) e la loro eventuale trasformazione (in input e in output).

Il beneficio dell'applicazione di un sistema di sviluppo di tale natura è quindi quello, da una parte, di gestire separatamente, sia in fase di sviluppo che in fase di manutenzione, le parti di software applicative, nelle quali e solo nelle quali risiede la logica del sistema informativo realizzato, dalle parti di software di sistema, dello sviluppo e della manutenzione delle quali sono responsabili un ristretto numero di persone, e d'altra parte di ottenere prodotti estremamente omogenei dal punto di vista formale, e quindi più facilmente manutenibili e aggiornabili, anche da persone che non hanno partecipato direttamente allo sviluppo.

## Architettura del sistema on-line

### Generalità

A prescindere dall'uso del Generatore di programmi che, si evidenzia, è uno strumento per aumentare la produttività nello sviluppo del software, garantendo 'automaticamente' il rispetto della maggior parte degli standard predisposti, per poter usufruire dei servizi che vengono resi disponibili dal sistema di sviluppo, le transazioni ed i programmi on-line dovranno rispondere alle regole che vengono illustrate nei seguenti paragrafi.

Regola base è che in ogni programma verrà gestito un solo pannello, nel senso che il pannello sarà preparato, inviato e saranno controllati e gestiti i dati e/o i comandi acquisiti con lo stesso. Questa regola da un lato semplifica la struttura dei programmi funzionanti in modalità pseudoconversazionale e dall'altro permette di ottenere una maggiore uniformità nei programmi stessi.

### Struttura delle transazioni conversazionali (CICS)

La modalità pseudoconversazionale disattiva la transazione quando è in attesa dell'input di dati da parte dell'utente, consentendo di abbassare notevolmente il carico del sistema.

Nella Figura 1 è illustrato schematicamente il flusso di una transazione.

Ad ogni transazione è associato, tramite la Program Control Table del CICS, sempre il programma ISYOTRNO, il quale svolge le funzioni di controllore delle transazioni. Il programma applicativo effettivo viene associato alla transazione tramite una tabella gestita dal sistema ISY.

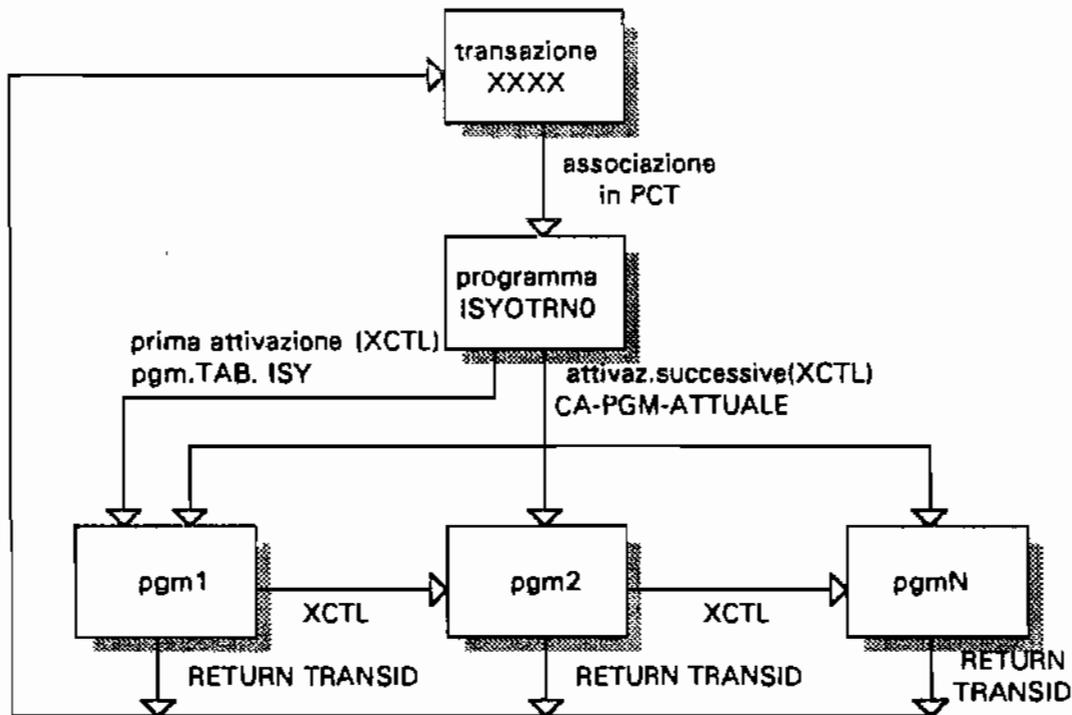


Figura 1 - Flusso tipico di una transazione

Al momento della prima attivazione ISYOTRNO verificherà l'abilitazione da parte dell'utente all'uso della transazione. Solo in caso di controllo positivo verrà ceduto il controllo al programma applicativo. Il programma ISYOTRNO provvederà inoltre ad inizializzare le aree di comunicazione di sistema.

La modalità pseudoconversazionale dei programmi applicativi comporta che, dopo aver effettuato la SEND di un pannello, il programma termina riattivando la transazione con un comando di RETURN TRANSID (xxxx).

### **Collegamento tra programmi**

La modalità pseudoconversazionale influenza anche la gestione delle comunicazioni tra un programma *chiamante* ed uno o più programmi *chiamati*, poichè, in realtà, l'attivazione di un sottoprogramma o il ritorno ad un programma chiamante si traducono in una chiusura del programma attivo per un passaggio di controllo ad un altro programma. Questo avviene con il comando XCTL che prevede due tipi di attivazione diversi se si deve o meno rientrare nel programma che cede il controllo. Nel primo caso si deve provvedere al salvataggio delle aree di lavoro, insieme al nome del chiamante, rispettivamente in una coda di temporary storage e in un apposito campo dell'area comune di sistema, da dove possano essere prelevati per la restituzione del controllo.

In questo contesto le modalità di chiamata di sottoprogrammi e di ritorno del controllo ai programmi chiamanti viene gestita tramite un sistema di livelli logici.

Questo sistema consiste nell'assegnare ad ogni programma un *numero di livello* al momento in cui viene attivato. Se un programma chiamato deve ritornare il controllo al

programma che lo ha attivato avrà un numero di livello progressivo rispetto al livello del chiamante, se, invece, il controllo non deve ritornare al chiamante il programma chiamato avrà lo stesso numero di livello del chiamante.

Prima di cedere il controllo, il programma provvede al salvataggio delle aree applicative in una coda di memoria (temporary storage in ambiente CICS) denominata tttt\$Snn, dove tttt identifica il codice terminale ed nn rappresenta il numero di livello (questo tipo di codifica serve per rendere univoca la coda di T.S.), da dove può, se riattivato, ripristinare i dati di interesse.

### **Acquisizione parametri di attivazione transazioni**

Poichè, in genere, ogni transazione ha bisogno di alcuni dati di attivazione, è stato realizzato un unico programma generalizzato per l'acquisizione di tali parametri.

Questo programma si chiama ISYOPAR0. Esso gestisce la preparazione del pannello di acquisizione dati ed il controllo formale sui dati digitati, secondo le modalità che verranno ora descritte.

Il programma di acquisizione utilizza un pannello generalizzato, la cui struttura è illustrata in Figura 2, costituito solo da campi senza valori iniziali. In particolare la prima riga (1) sarà riservata alla descrizione dell'elaborazione richiesta, tale descrizione sarà prelevata dall'archivio delle transazioni. L'ultima riga è riservata ai messaggi (2) e ad un campo (3) in cui possono essere indicati parametri associati a richieste di particolari funzioni. La riga precedente (4) contiene una breve descrizione delle Program Function attivabili da pannello. Le rimanenti righe sono destinate all'acquisizione dei parametri e sono suddivise ognuna in due zone. La parte a sinistra (5) di lunghezza pari a 18 caratteri è protetta e riservata ai testi di descrizione dei parametri. La parte rimanente (6), di lunghezza pari a 60 caratteri, è digitabile ed è la zona di specificazione dei parametri.

Il diagramma mostra un rettangolo con angoli arrotondati, diviso in una griglia di linee orizzontali e verticali. Sei punti di riferimento sono indicati con numeri in parentesi:

- (1) è posizionato nella parte superiore centrale.
- (2) è nella parte inferiore sinistra.
- (3) è nella parte inferiore destra.
- (4) è nella parte inferiore centrale.
- (5) è nella parte sinistra centrale.
- (6) è nella parte destra centrale.

Figura 2 - Layout pannello acquisizione dati di attivazione transazioni

Per ogni transazione, i dati di attivazione di una transazione possono provenire direttamente:

- associati al codice transazione,
- dal pannello di menù,
- da un altro programma (modalità *Navigazione*),
- tramite il pannello di acquisizione di Figura 2.

Il programma applicativo effettivo della transazione, che, alla prima attivazione, riceve eventualmente questi dati in DFHCOMMAREA, dopo i dati di sistema, verifica la completezza dei dati ricevuti, controlla cioè che questi siano sufficienti allo svolgimento della funzione, nel caso in cui questo non si verifichi attiva la richiesta degli stessi con eventuali indicazioni della incompletezza riscontrata. Per fare ciò il programma scrive in una coda di memoria, che si chiama `ttttSPnn`, i parametri di definizione dei dati che devono essere acquisiti (descrizione del campo, numero della riga sul pannello, controllo

da effettuare sul dato digitato) e passa il controllo a ISYOPARO che gestisce autonomamente l'invio del pannello, l'acquisizione dati ed i controlli formali e ritorna al programma chiamante per completare l'operazione con i controlli logici, passandogli i dati controllati in Coomunication Area.

Se i dati di attivazione sono più di uno, e se vengono immessi senza passare per il pannello di figura 2, essi devono essere forniti al programma associato alla transazione nell'esatto ordine in cui sono richiesti e ciascuno deve essere separato dal successivo da un punto esclamativo (!). Questa formattazione sarà a carico dell'utente nel caso in cui i dati giungano insieme con il codice transazione o dal pannello di menù, a carico del programma chiamante in modalità *Navigazione*, quando, cioè, una transazione è attivata con gli stessi dati della transazione di partenza.

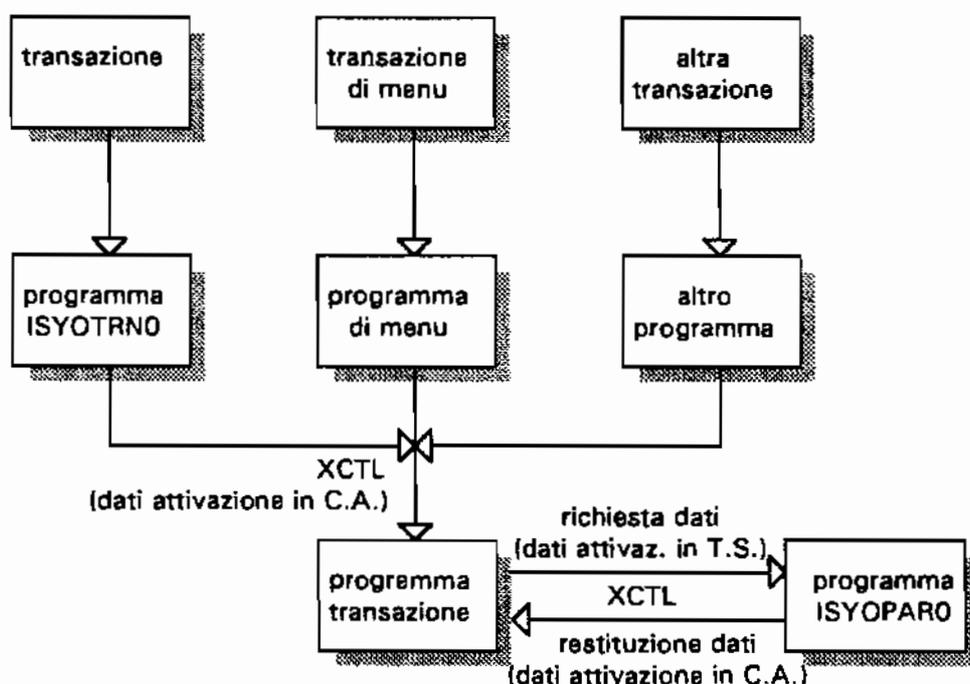


Figura 3 - Flusso tipico di attivazione per transazioni interattive

### **Visualizzazione liste e selezione dati**

Anche per la gestione di pannelli a lista è stato realizzato un programma generalizzato denominato ISYOVISO. Questo programma è in grado di gestire autonomamente la visualizzazione di un elenco omogeneo di dati, provvedendo allo scorrimento verticale dell'elenco stesso tramite la preparazione e l'invio dei pannelli contenenti tali dati, lasciando a carico del programma applicativo unicamente la predisposizione di tale elenco in una coda di memoria, il cui nome è `ttt$Vnn`, da cui il programma ISYOVISO possa prelevare i dati.

E' anche gestito un tasto funzione per richiedere la stampa on-line della lista dei dati presenti nella coda di memoria.

Da tali liste l'utente può, eventualmente, effettuare la selezione di una o più righe ed indicare la funzione che desidera eseguire. I dati introdotti vengono quindi passati al programma applicativo appositamente predisposto per eseguire la funzione (routine, programma, transazione) richiesta.

L'attivazione di questo programma può prevedere o meno il ritorno alla visualizzazione della lista al termine della esecuzione della funzione.

### **Tasti funzionali**

Il sistema ISY offre notevoli facilitazioni per quanto riguarda la gestione applicativa dei tasti funzionali.

La particolare modalità di definizione dei tasti funzionali li rende standardizzabili nell'ambito di ogni singola applicazione realizzata, in quanto l'associazione di un tasto alla funzione da svolgere viene fatta al di fuori dei programmi applicativi, dal programma di attivazione delle transazioni.

Nei programmi applicativi sono definiti solo gli accoppiamenti tasto-funzione gestiti dal programma stesso. Questo fatto, unito alla costruzione dinamica della riga dei tasti funzionali a cura di ciascun programma applicativo, fa sì che sui pannelli appaiano solo i tasti realmente attivabili.

### **Stampe on-line**

Le transazioni di stampa on-line sono, generalmente, gestite da due attivazioni dello stesso programma. La prima (associata al video richiedente) effettua l'acquisizione ed il controllo dei dati di attivazione tramite il colloquio con il programma di utilità ISYOPAR0 secondo le modalità già viste in precedenza, verificandone la validità formale e sostanziale.

Fatto questo, produce la seconda attivazione indirizzata sulla stampante, il cui codice è stato preventivamente definito per l'utente o da lui modificato, passando i dati di attivazione in Communication Area.

Al fine di omogeneizzare la scrittura di un programma di stampa on-line con quella di un programma di stampa batch, è stata realizzata l'utility ISYOPRNO, che si assume completamente il carico di gestire il buffer delle stampanti 3270. Questa routine viene richiamata all'interno dei programmi ogni volta che si vuole eseguire la stampa di una riga.

## Architettura del sistema batch

### Generalità

La struttura dei programmi batch in tale sistema di sviluppo non ha esigenze particolari rispetto a quelle di un normale programma strutturato. Si rimanda comunque al manuale citato in Introduzione per seguire una programmazione adeguatamente strutturata ed ottenere ugualmente programmi omogenei.

Il sistema mette comunque a disposizione il programma generalizzato per il controllo in input dei dati di attivazione e per la conversione in output dei dati da stampare ed un programma per il controllo della provenienza dei dati di input.

### Acquisizione parametri di attivazione

Il sistema prevede che i programmi debbano poter funzionare sia ricevendo l'input da normali schede parametro inserite nella Job Stream, sia da un archivio in cui siano stati preregistrati i dati di input o le richieste 'off-line' da elaborare. Per ottenere questo risultato è stata predisposta la routine ISYBIPMO che gestisce le seguenti funzioni:

- l'acquisizione dei parametri di elaborazione dalle varie fonti possibili
- la stampa dei parametri di elaborazione sulla stampante di sistema
- la stampa di un separatore con il nome dell'utente, per facilitare la distribuzione dell'output
- nel caso di programmi con chiamate SQL, l'inizializzazione dei parametri per effettuare la funzione di CONNECT al database relazionale.

Questo programma verrà guidato nella sua elaborazione da una scheda parametro presente nella Job Stream.

## La centralizzazione dei controlli

In tutte le precedenti esperienze si è dimostrata di importanza fondamentale una funzione centralizzata di controllo e conversione dei dati. Tale funzione permette innanzi tutto di garantire la omogeneità di tutti i controlli effettuati sullo stesso dato e poi di poter disporre facilmente del dato in formato editato.

Queste tecniche si basano sull'esistenza di un unico programma al quale viene fornito il dato da controllare od editare insieme con le sue caratteristiche. In base a queste caratteristiche il programma effettua il controllo, restituendo informazioni circa il suo risultato, o la conversione, restituendo il suo valore *esterno* o *interno* nei casi rispettivamente di output e di input.

Un unico sorgente ISYxCCD, che diventa ISYOCCD in ambiente on-line, e ISYBCCD in ambiente batch espleta questa funzione all'interno del sistema di sviluppo.

### **Il controllo ed edit dei campi in on-line**

La modalità di attivazione prevista è tramite LINK in quanto ha una funzionalità simile a quella di una routine.

L' area di interfaccia (copy ISYCCCD) viene passata in DFHCOMMAREA. I campi di ingresso e quelli di uscita della funzione sono contenuti all'interno di questa interfaccia.

La modalità normale di chiamata è quella che prevede un controllo da pannello, per cui, in caso di assenza di errori, restituisce sul campo di ingresso il dato in formato editato tale da poter essere di nuovo portato su pannello; nel caso di ingresso a LOW-VALUE (cioè dato non digitato) il programma provvede a riformare il pannello editando il dato dal campo interno. In questa modalità deve essere quindi fornito come dato di ingresso non solo il dato esterno, ma anche il dato in formato interno già acquisito o presente sul record (nell'apposito campo di interfaccia dedicato alla funzione).

### **Il controllo ed edit dei campi in batch**

Il programma di controllo e conversione dati ISYBCCD0 presenta le stesse funzionalità previste per l'on-line. Anche la sua modalità di attivazione richiama le funzionalità on-line in quanto viene caricato dinamicamente in memoria al momento della prima richiesta. Questa modalità permette quindi, in caso di modifica di ISYBCCD0, di far lavorare automaticamente tutti i programmi con la nuova versione senza dover procedere ad una nuova fase di link-edit e peraltro senza ingigantire inutilmente i programmi.

In ambiente VSE, la richiesta di servizi ad ISYBCCD0 viene effettuata tramite la routine ISYRCCD0 che ne richiede il caricamento in memoria cedendogli il controllo. Viene cioè effettuato un link-edit dinamico di ISYBCCD0 al programma chiamante.

L' interfaccia di comunicazione fra il programma applicativo ed ISYBCCD0 è contenuta nel copy ISYCCCD e viene trasmessa in Linkage Section secondo le modalità normali per sottoprogrammi richiamati con CALL.

Per semplificare il riempimento dell'interfaccia è possibile sfruttare la possibilità offerta nel solo ambiente batch di poter passare i dati in formato interno direttamente fornendo il campo come parametro della CALL. Per permettere al programma ISYBCCD0 di distinguere questa modalità di passaggio dati rispetto al riempimento completo dell'interfaccia sono previste due opzioni per richiedere il controllo di un dato o il suo edit.

## **I servizi per l'utente**

Insieme con il sistema di sviluppo sono state realizzate applicazioni accessorie di sistema che possono essere normalmente integrate col sistema informativo realizzato completando e arricchendo le funzionalità proprie di ciascuna applicazione.

Tali applicazioni sono strettamente collegate con il sistema di sviluppo nel senso che alcune funzioni sono ottenute grazie alla particolare architettura impiegata.

### **Security**

Ogni volta che un utente attiva una transazione, viene automaticamente controllato che sia stata eseguita la procedura di sign-on e che l'utente sia abilitato all'uso della transazione attivata.

Il sign-on può essere indifferentemente quello del sistema TP-monitor (CICS) dell'host o quello del sistema di sviluppo, nei casi in cui il livello di aggiornamento del software di base del sistema CICS lo permetta, altrimenti occorre effettuare la doppia sign-on prima del CICS e poi del sistema di sviluppo ISY.

In ogni caso il sistema ISY deve conoscere il nome dell'utente che ha iniziato la sessione di lavoro.

In relazione alla security sono presenti tre archivi che contengono rispettivamente dati relativi all'utente, alle transazioni e alle abilitazioni.

Il programma ISYOSON0 (attualmente nominato ISYO0000) acquisisce i dati dal sign-on effettuato, controlla le abilitazioni all'interno del sistema ISY e permette l'accesso al sistema se le abilitazioni esistono.

Le abilitazioni contengono anche delle opzioni che servono a specificare il tipo di accesso ai dati mediante le transazioni abilitate: solo lettura, lettura e aggiornamento, ecc.

Ogni tentativo di accesso errato viene conteggiato dal sistema. Al completamento di un sign-on corretto, vengono fornite informazioni sul sign-on precedente e sugli eventuali tentativi abortiti di sign-on con quello stesso nominativo utente.

Con i dati dell'utente presenti nella coda associata al terminale, viene controllata l'abilitazione all'uso di ciascuna delle transazioni attivate.

Tutti i sign-on e sign-off effettuati vengono registrati anche su un archivio apposito per permettere eventuali funzioni di auditing.

### **Help**

Dall'ambiente interattivo l'utente può consultare la documentazione (help) che permette di conoscere le modalità di utilizzazione delle transazioni cui è abilitato.

Si possono distinguere i seguenti tipi di help:

**Help di menu**

Consiste nella descrizione generale dei sottomenu e/o transazioni definite da un determinato menu. A seconda dell'organizzazione dei menu, questo help può essere utilizzato molto efficacemente per documentare complessivamente le procedure comprendenti più transazioni o gruppi di transazioni. Viene richiesto dall'utente dal menu, selezionando con 'h' il sottomenu voluto.

**Help di transazione**

Consiste nella descrizione generale delle funzionalità ottenibili con la transazione xxxx. Viene richiesto dall'utente mediante la pressione del tasto di help durante l'esecuzione di una transazione oppure mediante la transazione di help seguita dal codice transazione oppure dal menu, selezionando con 'h' la transazione voluta.

**Help di pannello**

Consiste nella presentazione, quando è significativa, di una descrizione più tecnica della transazione con particolare riferimento ai campi del pannello.

**Help di campo**

Se richiesto con l'introduzione di un ? sul campo e Invio si apre una finestra con informazioni sui controlli che vengono effettuati sullo stesso. Per campi tabellari viene fornito l'elenco dei valori accettabili da cui è possibile la selezione diretta. È importante considerare che tutte tali informazioni sono prelevate direttamente dalla stessa fonte utilizzata dalle routine che effettuano i controlli sui dati introdotti in input, garantendo così l'allineamento continuo e automatico della documentazione.

Se richiesto con la pressione del tasto di help e il cursore posizionato sotto il campo stesso, viene aperta una finestra con la documentazione del campo che è stata caricata nel dizionario dati.

**Help di messaggio**

Consiste nella spiegazione delle segnalazioni e dei messaggi di errore ricevuti e contiene, ove necessario, una descrizione degli interventi da effettuare per correggere l'errore. E' richiedibile posizionando il cursore sull'ultima riga del pannello e premendo il tasto di help.

**Help di lista**

Consiste nella visualizzazione dell'elenco dei caratteri di selezione utilizzabili dai pannelli a lista e cioè delle funzioni attivabili su uno o più elementi di una lista

Anche in questo caso l'help può essere attivato dall'utente inserendo un carattere ? sul campo di input del pannello.

## **Menu**

Il sistema prevede un menù delle transazioni strutturato in livelli gerarchici, che elencano successivamente aree e sottoaree (sottomenu) applicative, e infine le transazioni cui l'utente è abilitato.

Tali aree, secondo le necessità organizzative degli utenti, spesso possono raggruppare le transazioni per procedure operative.

Queste strutture gerarchiche possono essere costruite dinamicamente.

Ad un singolo utente, nella fase della sua introduzione del sistema, viene assegnata una di queste strutture. Quando l'utente effettua un sign-on viene mostrato automaticamente il menu che gli è stato associato.

## **Dizionario dati**

Il dizionario dati contiene la documentazione tecnica e di utente di tutti i campi degli archivi dell'applicazione. A tale documentazione accedono i programmi di sistema per i seguenti principali scopi:

- controllo automatico formale dei dati immessi nel sistema in base alla tipologia e alle caratteristiche del campo stesso
- editing del campo su pannelli e su stampe in funzione delle caratteristiche del campo
- help on-line a livello di campo richiesto dall'utente.

A tale documentazione dei dati accede anche il sistema IDS che può essere utilizzato per configurare la struttura dei componenti dell'intero progetto.

## Richieste off-line

La gestione delle richieste off-line consiste in una serie di moduli che consentono all'utente di una applicazione di fornire una richiesta di elaborazione con i relativi parametri ed all'operatore di schedulare l'esecuzione dell'elaborazione in forma semplificata.

Questa funzionalità consta di tre moduli:

- modulo di acquisizione della richiesta di elaborazione;
- modulo di schedulazione delle elaborazioni richieste;
- modulo di passaggio dei parametri al programma applicativo.

Il modulo di acquisizione della richiesta di elaborazione fornisce l'interfaccia per l'utente. È quindi basato sull'invio di un pannello con la richiesta dei dati specifici dell'elaborazione, con la loro acquisizione e registrazione in un "deposito" che in ambiente CICS sarà un file VSAM. L'acquisizione dei parametri di elaborazione viene effettuata tramite un programma generalizzato con la stessa tecnica già vista in precedenza per l'acquisizione dei dati di attivazione delle transazioni. Il programma generalizzato effettua esso stesso i controlli di validità formale dei dati e gestisce le segnalazioni di errore. Solo se non esistono errori provvede alla registrazione dei dati sul file delle richieste off-line. Sul file delle richieste off-line ogni richiesta è identificata dal codice di transazione, dal codice dell'utente che ha effettuato la richiesta e dalla data in cui è stata effettuata la richiesta stessa. Il programma generalizzato può essere riattivato per permettere la correzione delle richieste già registrate.

Il modulo di schedulazione delle elaborazioni richieste permette all'operatore di verificare le richieste di elaborazione effettuate dagli utenti, di visualizzare i loro parametri di attivazione e di rilasciarle per l'elaborazione.

Può essere visualizzato l'elenco delle richieste di elaborazione, può essere permessa la selezione di una richiesta per la visualizzazione ed attivato il rilascio singolo o per gruppi.

I programmi applicativi, per poter utilizzare i parametri inseriti dagli utenti, devono accedere al file degli off-line per prelevarli. È comunque garantita la possibilità di far girare i programmi in forma batch nativa, cioè inserendo le schede parametro, con lo stesso formato, direttamente nella job stream.

Il modulo di passaggio dei parametri ai programmi applicativi è stato realizzato per provvedere a queste funzionalità. Si tratta, in pratica, di una routine generalizzata che acquisisce i parametri dalle varie fonti, fornendoli sempre in modo uniforme al programma applicativo. Questa routine basa il suo modo di lavorare su una scheda speciale ad esso destinata inserita nella job stream come prima scheda parametro. In questa scheda sono fornite le indicazioni della fonte da cui i parametri devono essere prelevati. Questa routine fornisce ulteriori funzionalità quali la stampa di un separatore (opzionale) con l'indicazione dell'utente richiedente e la possibilità di memorizzare i parametri su uno speciale archivio per poter essere utilizzati da altre fasi costituenti la stessa job stream. Nel caso di elaborazione proveniente da file di off-line questa stessa routine provvederà a gestire un indicatore di richiesta elaborata presente sul file stesso.