



MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA LOTTA ALLA CONTRAFFAZIONE
UFFICIO ITALIANO BREVETTI E MARCHI

DOMANDA NUMERO	101998900699258
Data Deposito	21/08/1998
Data Pubblicazione	21/02/2000

Priorità	08/920773
Nazione Priorità	US
Data Deposito Priorità	

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F		

Titolo

PROCEDIMENTO PER L'INSTALLAZIONE E LA VERIFICA DEL SOFTWARE PER UN SISTEMA DI COMPUTER PRODOTTO SU ORDINAZIONE.

Descrizione dell'invenzione industriale dal titolo:
"Procedimento per l'installazione e la verifica
del software per un sistema di computer prodotto su
ordinazione"

di: Dell Computer Corporation, nazionalità
statunitense, One Dell Way, Round Rock, Texas
78682-2244 U.S.A.

Inventori designati: Richard D. Amberg, Roger W.
Wong, Michael A. Brundridge

Depositata il: 21 AGOSTO 1998

TO 98A 000720

SFONDO DELL'INVENZIONE

Riferimento incrociato a domande correlate

La presente domanda è relativa alla domanda di
brevetto statunitense col numero di serie _____
anch'essa pendente, n. di protocollo N-5022 US,
depositata lo stesso giorno della presente domanda,
intitolata Software Installation And Testing For A
Build-To-Order Computer System e che nomina Richard
D. Amberg, Roger W. Wong e Michael A. Brundridge
come inventori, la domanda essendo qui incorporata
per riferimento nella sua interezza.

La presente domanda è relativa alla domanda di
brevetto statunitense col numero di serie _____
anch'essa pendente, n. di protocollo M-5216 US,

BUZZI, NOTARO &
ANTONIELLI D'OUIX
S.r.l.

depositata lo stesso giorno della presente domanda, intitolata Database For Facilitating Software Installation And Testing For A Build-To-Order Computer System e che nomina Richard D. Amberg, Roger W. Wong, and Michael A. Brundridge come inventori, l'applicazione essendo qui incorporata per riferimento nella sua interezza.

Campo dell'invenzione

La presente forma di attuazione è relativa alla diagnostica di sistemi di computer e più in particolare ad un procedimento per sequenziare le fasi di installazione e/o verifica del software per un sistema di computer.

Descrizione della tecnica anteriore

I sistemi di personal computer in generale e sistemi di personal computer IBM compatibili in particolare hanno ottenuto un utilizzo ampio per fornire potenza di calcolo a molti segmenti della società. Un sistema di personal computer può essere definito di solito come microcomputer da tavolo, da pavimento o portatile che comprende una unità di sistema avente un processore di sistema ed una memoria volatile e non volatile associata, un monitor di visualizzazione, una tastiera, una o più unità a dischetto, un dispositivo di memorizzazione

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

a disco fisso ed una stampante opzionale.

E' noto installare software o eseguire verifiche su sistemi di computer prima che essi siano spediti alle aziende o ai singoli clienti. L'obiettivo dell'installazione e della verifica del software è quello di produrre in modo efficiente un sistema di computer utile e affidabile che può essere inviato alle aziende e alle persone privo di errori e pronto per il funzionamento. In generale, la verifica rileva e analizza gli errori che hanno luogo sia nelle porzioni hardware sia nelle porzioni software del sistema di computer. Un elenco parziale dei test hardware del sistema di computer potrebbe comprendere una diagnostica sui componenti hardware quali un processore, una memoria, un dispositivo di memorizzazione a disco, un dispositivo audio, un dispositivo grafico, una tastiera, un mouse ed una stampante. L'installazione software comprende spesso il caricamento di un pacchetto desiderato di software sul sistema di computer, la preparazione di variabili di ambiente appropriate per il computer, e la preparazione di file di inizializzazione appropriati per il software caricato. La verifica del software comprende spesso l'accertamento che

BUZZI, NOTARO &
ANTONIELLI D'OUX
s. r. l.

una versione desiderata del software è stata installata sul sistema di computer e che sono presenti drive appropriati sul sistema di computer.

E' noto nell'industria installare software e testare i sistemi di computer durante la fabbricazione eseguendo una procedura fissa prima che essi siano spediti ai clienti. Per esempio, si crea un dischetto contenente certi test diagnostici per un certo tipo di sistema di computer. Il dischetto comprende file batch lunghi e spesso complicati che dirigono i processi di installazione diagnostica del software. Il dischetto contiene inoltre tutti i file eseguibili per eseguire test sul sistema di computer acquistato.

Ogni sistema di computer che viene costruito è dotato di una rispettiva copia di questo dischetto. Questi dischetti accompagnano il sistema di computer che viene costruito per tutta la fabbrica durante il processo di fabbricazione, test venendo eseguiti sul rispettivo sistema di computer secondo l'ordine inerente nel file batch. Se occorre effettuare una modifica al processo, il file batch viene modificato in modo corrispondente aggiungendo o togliendo porzioni dal batch. Questa modifica nel file batch dà come risultato una corrispondente

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

modifica dei parametri di test (compresa la sequenza in cui i test sono eseguiti) di ogni successivo sistema di computer in fabbricazione, dato che ogni sistema di computer condivide la stessa procedura diagnostica di file batch.

Mentre le soluzioni diagnostiche di questo tipo hanno presentato un certo grado di utilità nell'aumentare l'affidabilità dei sistemi di computer prima della spedizione, rimane spazio per perfezionamenti. Per esempio, dato che le verifiche continuano a diventare sempre più complicate e approfondite, i file batch e i file eseguibili dei test diagnostici superano spesso le capacità di memorizzazione di un dischetto. Inoltre, è spesso difficile e impossibile personalizzare le procedure di verifica e di installazione del software per un singolo sistema di computer fabbricato su ordinazione o per una certa famiglia di sistemi di computer senza modificare le verifiche per altri sistemi o famiglie. Inoltre, è difficile o impossibile modificare l'ordine di installazione o verifiche del software per un sistema di computer singolo costruito su ordinazione o per una certa famiglia di sistemi di computer senza modificare l'ordine per gli altri sistemi e famiglie. Infine,

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

la natura spesso complicata delle attuali strutture di file batch rende talvolta difficile per i costruttori risolvere i problemi o mantenere le procedure di verifica o installazione del software in maniera rapida ed efficiente. In modo corrispondente, sarebbe desiderabile prevedere un procedimento perfezionato per installare software e sottoporre a verifica i sistemi di computer prima che essi siano spediti ai clienti.

L'INVENZIONE

Un procedimento per installare e/o verificare software per un sistema di computer prodotto su ordinazione avente una pluralità di componenti comprende una pluralità di elementi. Il sistema di computer ha una sequenza di fasi associata. La sequenza di fasi comprende una pluralità di fasi dove ogni fase comprende almeno un comando ed una fase associata ad un rispettivo descrittore di componente. Un descrittore di componente corrisponde ad un rispettivo componente del sistema di computer. Il procedimento comprende le fasi di accedere ad un comando all'interno della sequenza di fasi; creare un file comprendente un indicazione di inizio dell'esecuzione; eseguire il comando; cancellare il file comprendente l'indicazione di

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

inizio di esecuzione quando il comando completa l'esecuzione; e ripetere le fasi di accedere, creare ed eseguire e cancellare per la pluralità di fasi.

In una forma di attuazione preferita, un procedimento per installare e/o verificare software per un sistema di computer prodotto su ordinazione avente una pluralità di componenti comprende una pluralità di elementi. Il sistema di computer ha una sequenza di fasi associata. La sequenza di fasi comprende una pluralità di fasi dove ogni fase comprende almeno un comando ed una fase associata ad un rispettivo descrittore di componente. Un descrittore di componente corrisponde ad un rispettivo componente del sistema di computer. Il procedimento comprende le fasi di leggere un comando dalla sequenza di fasi; ricercare il comando in una unità locale del sistema di computer; eseguire il comando dall'unità locale se il comando è memorizzato sull'unità locale del sistema di computer; eseguire il comando da un server se il comando è memorizzato su una unità locale del sistema di computer.

In un'altra forma di attuazione preferita, un procedimento per installare e/o verificare software

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

per un sistema di computer prodotto su ordinazione avente una pluralità di componenti comprende una pluralità di elementi. Il sistema di computer ha una sequenza di fasi associata. La sequenza di fasi comprende una pluralità di fasi dove ogni fase contiene almeno un comando ed una fase associata ad un rispettivo descrittore di componente. Un descrittore di componente corrisponde ad un rispettivo componente del sistema di computer. Il procedimento comprende le fasi di leggere un comando dalla sequenza di fasi; eseguire il comando sul sistema di computer; creare un indicatore temporale di inizio che indica quando è iniziata l'esecuzione del comando; e creare un indicatore temporale di fine che indica quando è terminata l'esecuzione del comando.

Il procedimento descritto fornisce così una installazione del software e una verifica del computer efficaci che consentono una immediata risoluzione degli errori e una personalizzazione dei sistemi di computer prodotti su ordinazione. Il progetto modulare del procedimento consente in modo vantaggioso il mantenimento elementare di un sistema di verifica e la rapida creazione di fasi per nuovi sistemi e famiglie di computer.

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

Questi ed altri scopi, caratteristiche e vantaggi dell'invenzione saranno ulteriormente descritti e risulteranno più immediatamente chiari da una revisione della descrizione dettagliata delle forme di attuazione preferite che segue.

BREVE DESCRIZIONE DEI DISEGNI

La Figura 1 è un diagramma schematico che illustra l'installazione di software e la verifica.

La Figura 2 è un diagramma schematico dell'installazione del software e la verifica secondo un'altra forma di attuazione.

La Figura 3A è un diagramma di flusso per convertire un ordine di computer in un record descrittore di sistema secondo la presente invenzione.

La Figura 3B illustra una porzione di un ordine di computer esemplificativo, il file di Record di Montaggio di Base (BAR), e record descrittore di sistema.

La Figura 4 è un diagramma di flusso per creare e fornire una sequenza di fasi.

La Figura 5 è un diagramma di flusso più dettagliato per creare una sequenza di fasi.

BUZZI, NOTARO &
ANTONIELLI D'OUIX
s.r.l.

La Figura 6 illustra una struttura di una base dati.

La Figura 7 è un diagramma di flusso per modificare un record descrittore di sistema e la sequenza di fasi.

La Figura 8 illustra un esempio di un file di fasi prima di essere eseguito.

La Figura 9 illustra il file di fasi esemplificativo dopo essere stato eseguito.

La Figura 10 è un diagramma di flusso del funzionamento di un programma per rivedere una sequenza di fasi.

La Figura 11 è un diagramma di flusso più dettagliato del funzionamento del programma di Figura 10 per eseguire una sequenza di fasi.

DESCRIZIONE DETTAGLIATA

Quanto segue evidenzia una descrizione dettagliata del modo migliore previsto per realizzare l'invenzione. La descrizione vuole essere illustrativa e non deve essere presa come limitativa. Nei disegni, gli elementi analoghi e simili possono essere designati dallo stesso numero di riferimento. Nella descrizione, un modulo è definito come comando o insieme di comandi.

La Figura 1 è un diagramma schematico del sistema di installazione del software e di verifiche 90. Durante il funzionamento, si colloca un ordine 92 per acquistare un sistema di computer obiettivo 160 prodotto su ordinazione. Il sistema obiettivo 160 deve essere fabbricato in modo da contenere una pluralità di componenti hardware e software. Per esempio, il sistema obiettivo 160 potrebbe comprendere una certa marca di unità a disco rigido, un particolare tipo di monitor, una certa marca di processore, ed una particolare versione di un sistema operativo. Prima che il sistema obiettivo 160 sia spedito al cliente, la pluralità di componenti vengono installati e verificati. Tale installazione del software e tali verifiche garantiscono in modo vantaggioso un sistema di computer operante ed affidabile che è pronto a funzionare dopo essere ricevuto.

Poiché famiglie diverse di sistemi di computer e singoli componenti di computer diversi richiedono fasi di installazione del software e di verifica diverse, è necessario determinare quali test devono essere eseguiti sul sistema obiettivo 160 e in quale ordine questi test dovrebbero essere eseguiti in modo da ottenere un processo di installazione

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

del software e di verifiche efficace. Il dispositivo di realizzazione di fasi 140' è un sistema di computer configurato in modo da sequenziare le fasi di installazione del software e di verifica da eseguire sul sistema obiettivo 160. Per sequenziare le fasi di installazione del software e/o verifiche, il dispositivo di realizzazione di fasi 140, e più in particolare il programma di sequenziamento 204 che risiede sul dispositivo di realizzazione di fasi 140, legge dapprima una pluralità di descrizione di componente dal file descrittore 96. Il file descrittore 96 è previsto convertendo un ordine 92, che corrisponde ad un sistema di computer desiderato che ha componenti desiderati, in un formato leggibile da computer tramite il modulo di conversione 94.

I descrittori di componente sono descrizioni leggibili da computer del componente del sistema obiettivo 160 i quali componenti sono definiti dall'ordine 92. Nella forma di attuazione preferita, i descrittori di componente sono compresi in un file descrittore chiamato record descrittore di sistema che è un file leggibile da computer contenente un elenco dei componenti, i componenti hardware e/o software, che devono essere

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

installati sul sistema obiettivo 160. Avendo letto la pluralità di descrittori di componente, il programma di sequenziamento 204 reperisce una pluralità di fasi di installazione del software e/o di verifica corrispondenti ai descrittori di componente dalla base dati 100 sulla connessione di rete 110. La connessione di rete 110 può essere qualsiasi connessione di rete ben nota nella tecnica, quale una rete di area locale, un intranet, o l'internet. Informazioni contenute nella banca dati 100 possono essere aggiornate tramite una modifica illustrata dalla freccia 130.

Avendo reperito le fasi di installazione del software e/o di verifica appropriate per il sistema obiettivo 160, il programma di sequenziamento 204 sequenzia le fasi in un predeterminato ordine secondo i numeri di sequenza corrispondenti ad ogni fase. Avendo sequenziato le fasi richieste per il sistema obiettivo 160, il programma di sequenziamento 204 scrive una serie di file di uscita sul disco di fasi 150. Nella forma di attuazione evidenziata in Figura 1, i file di uscita comprendono file di testo contenenti linee di comando appropriate per eseguire le fasi di installazione del software e/o di verifica

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

appropriate sul sistema obiettivo 160. L'esecuzione è eseguita nel predeterminato ordine secondo i numeri di sequenza corrispondenti ad ogni fase. Il disco di fase 150 accompagna il sistema obiettivo 160 nella fabbrica dove i test vengono eseguiti direttamente dal disco di fasi 150 oppure, in alternativa, dal file server 190, connesso al sistema obiettivo 160 tramite la connessione di rete 180. Preferibilmente, la connessione di rete 180 è un dispositivo di rete generico connesso ad una corrispondente porta di rete del sistema di computer obiettivo. A seguito dell'esecuzione delle fasi di installazione del software e di verifica, i risultati dell'installazione e dei test sono registrati di nuovo nel file server 190 sulla connessione di rete 180.

La Figura 2 è un diagramma schematico del sistema di installazione del software e di verifiche 192 in conformità con un'altra forma di attuazione della presente invenzione. Un cliente colloca un ordine 92 per acquistare un sistema di computer obiettivo 160 prodotto su ordinazione. Il sistema obiettivo 160 deve essere fabbricato in modo da contenere una pluralità di componenti i quali componenti possono comprendere componenti sia

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

hardware sia/o software. Prima che il sistema obiettivo 160 sia spedito al cliente, la pluralità di componenti vengono installati e testati. Tale installazione e verifica garantisce in modo vantaggioso un sistema di computer operante ed affidabile che è pronto a funzionare quando viene ricevuto dal cliente.

Per sequenziare le fasi di installazione del software e di verifica, il programma di sequenziamento 204 legge una pluralità di descrittori di componente dal file descrittore 96. L'ordine 92 viene convertito nel file descrittore 96 tramite il modulo di conversione 94. I descrittori di componente sono descrizioni leggibili da computer del componente da sistema obiettivo 160. Nella forma di attuazione preferita, i descrittori di componente sono compresi in un file descrittore chiamato record descrittore di sistema, un file leggibile da computer contenente un elenco di ogni componente, sia hardware sia software, che deve essere installato sul sistema obiettivo 160. Il record descrittore di sistema può essere memorizzato direttamente sul file server 202. Il programma di sequenziamento 204 reperisce una pluralità di fasi di installazione del software

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

e/o di verifica corrispondente ai descrittori di componente dalla base dati 100. Avendo reperito le fasi di installazione del software e/o di verifiche appropriate per il sistema obiettivo 160, il programma di sequenziamento 204 sequenza le fasi in un predeterminato ordine secondo i numeri di sequenza corrispondenti ad ogni fase. Avendo sequenziato le fasi richieste per il sistema obiettivo 160, il programma di sequenziamento 204 dirige l'esecuzione delle fasi di installazione del software e di verifica sul sistema obiettivo 160 nell'ordine predeterminato tramite le connessioni di rete 195 e 180. Si desidera che la connessione di rete 200 sia un dispositivo di rete generico collegato ad una corrispondente porta del sistema obiettivo 160. La rete 195 può essere qualsiasi connessione di comunicazione ben nota nella tecnica. A seguito dell'esecuzione delle fasi di installazione del software e/o di verifica, i risultati dell'installazione e dei test sono registrati di nuovo nel file server 202 sulla connessione di rete 200 oppure sono memorizzati all'interno di una base dati appropriata. Come appare chiaro dall'illustrazione, non si ha alcuna necessità di un sistema di computer di

BUZZI, NOTARO &
ANTONIELLI D'OUIX
s.r.l.

realizzazione delle fasi 140 separato come in Figura 1. In aggiunta, il disco di fasi 150 non è necessario. Piuttosto, è necessario soltanto il disco di boot 220, che è configurato in modo da inizializzare il sistema obiettivo 160, per accompagnare il sistema obiettivo 160 in fabbrica.

Avendo descritto in generale i sistemi di installazione del software e di verifica, si presterà ora attenzione alla descrizione del funzionamento dei sistemi evidenziati nelle Figure 1 e 2 in maggior dettaglio.

La Figura 3A illustra il processo preferito in cui un ordine per un sistema di computer è convertito in un record descrittore di sistema leggibile da computer. Più specificamente, nell'elemento 300, si riceve un ordine per un sistema di computer obiettivo. Questo ordine può essere in una qualsiasi di un numero innumerevole di forme. Per esempio, sono possibili diversi formati di ordinamento così come diversi meccanismi di consegna di ordini: Per esempio, gli ordini per un sistema di computer obiettivo possono essere ricevuti per telefono, per posta, o su reti di computer (ad esempio sull'internet). Indipendentemente dai mezzi per la ricezione o la

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

forma dell'ordine, l'ordine comprende il tipo di sistema di computer obiettivo che un cliente desidera acquistare e, eventualmente, un elenco esplicito dei particolari componenti che il cliente desidera che il sistema di computer obiettivo comprenda. Dopo che si riceve l'ordine, il controllo opera una transizione al modulo di trasmissione 310 durante la quale l'ordine di sistema di computer obiettivo viene trasmesso su una rete di computer ad un sistema di fabbricazione (non illustrato) che produce il sistema di computer obiettivo. L'ordine di sistema di computer obiettivo viene anche fornito al sistema di installazione del software e di verifica dove viene incanalato in un programma di conversione nel modulo 320. La rete di computer utilizzata nel modulo 310 può essere di qualsiasi tipo ben noto nella tecnica.

Il programma di conversione converte l'ordine di sistema di computer obiettivo in un record utile per il processo di fabbricazione. Più specificamente, il programma di conversione converte l'ordine di computer dapprima in un record chiamato un file BAR nel modulo 330. Preferibilmente, il file BAR contiene un

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

identificatore unico che identifica lo specifico sistema di computer obiettivo in fabbricazione. Il file BAR contiene inoltre un elenco dettagliato dei componenti, che possono comprendere sia hardware sia software, da inserire nel sistema obiettivo. Inoltre, si desidera che il file BAR contenga numeri categorici specifici per costruttore o altri identificatori utili per ogni componente. Infine, il file BAR può contenere informazioni specifiche del cliente quali nome, indirizzo e numero telefonico.

A seguito della creazione del file BAR nel modulo 330, si crea un record descrittore di sistema nel modulo 340. Un record descrittore di sistema, nella forma di attuazione preferita, è un file leggibile da computer che è descrittivo dei componenti hardware e software da inserire nel sistema di computer obiettivo. In una forma di attuazione preferita, il record descrittore di sistema contiene un elemento di componenti del sistema obiettivo in un formato comprendente etichette hardware, etichette software, etichette di informazioni e commenti. Una etichetta hardware identifica al programma di sequenziamento 204 che le informazioni che seguono l'etichetta sono

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

relative ad un componente hardware. Analogamente, l'etichetta software identifica le informazioni che seguono l'etichetta come relative ad un componente software. L'etichetta di informazioni indica che seguiranno informazioni generali. I commenti consentono di inserire varie frasi nel record descrittore di sistema che sono ignorate dal programma di sequenziamento 204. E' desiderabile che il record descrittore di sistema sia un file di testo che sia leggibile dalle persone e facile da capire. Tale file consente in modo vantaggioso una facile risoluzione degli errori e una facile manutenzione del processo di installazione e di verifica. Si potrà apprezzare che il record descrittore di sistema potrebbe essere qualsiasi elenco di identificatori unici che corrispondono ad un insieme unico di soggetti, per esempio, in un esempio semplice, il record descrittore di sistema può essere un elenco di numeri categorici.

La Figura 3B illustra un ordine esemplificativo 350 di un sistema di computer obiettivo, un corrispondente file BAR 360, ed un corrispondente record descrittore di sistema 370. L'ordine di sistema di computer obiettivo 350 contiene il nome di una famiglia di computer, in

questa illustrazione, la famiglia "X". Sono inoltre compresi nell'ordine di sistema di computer obiettivo 350 tre componenti hardware esemplificativi comprendenti un processore Pentium®, un'unità a disco rigido ed un monitor. Il file BAR 360 deriva dall'esecuzione dell'ordine del sistema di computer obiettivo 350 attraverso un programma di conversione come illustrato nel modulo 320 di Figura 3A. Il file BAR 360 contiene un identificatore unico per il sistema di computer obiettivo specifico nell'ambito della famiglia X. Il file BAR 360 comprende inoltre i numeri categorici specifici per costruttore per ciascuno dei componenti elencati nell'ordine di sistema di computer obiettivo. Inoltre, il file BAR 360 contiene un identificatore che indica la quantità desiderata di ogni componente così come una descrizione di testo di ogni componente da inserire sul sistema di computer obiettivo. Il sistema 90 utilizza il file BAR 360 per creare il record descrittore di sistema 370.

Come illustrato, il record descrittore di sistema 370 contiene anche l'identificatore unico per lo specifico sistema di computer obiettivo all'interno della famiglia X. Inoltre, il record

descrittore di sistema 370 contiene etichette appropriate, che indicano qui che il processore, l'unità a disco rigido e il monitor sono tutti componenti hardware, piuttosto che software. Il record descrittore di sistema 370 descrive questi componenti in una descrizione di test. In aggiunta, il record descrittore di sistema 370 esemplificativo contiene una etichetta software che indica che occorre installare o verificare un certo software sul sistema di computer obiettivo appartenente alla famiglia X. Per esempio, l'etichetta software potrebbe indicare che un certo sistema operativo appropriato per il processore Pentium® sia sempre installato sull'unità a disco rigido del sistema di computer obiettivo appartenente alla famiglia X.

In Figura 4, è evidenziato il procedimento generale preferito per sequenziare le fasi di installazione del software e verifica. Nel modulo 400, si genera l'identificatore unico del sistema di computer obiettivo per il sistema di computer obiettivo 160. Nella forma di attuazione illustrata in Figura 1, un utente seduto al sistema di computer di realizzazione delle fasi 140 fornisce l'identificatore unico (ad esempio l'identificatore

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

BAR che funziona come codice di rintracciamento) nel programma di sequenziamento 204 del dispositivo di realizzazione di fase 140. In alternativa, nella forma di attuazione di Figura 2, l'identificatore unico viene letto automaticamente nel programma di sequenziamento 204 dopo che si riceve l'ordine di sistema di computer obiettivo.

Nel modulo 410, è collocato un record descrittore di sistema corrispondente all'identificare BAR. Nella forma di attuazione di Figura 1, la connessione di rete 110 o la connessione di rete 195 localizza il record descrittore di sistema. Nella forma di attuazione di Figura 2, la connessione di rete 195 localizza il record descrittore di sistema. Nel modulo 420, il record descrittore di sistema localizzato è fornito al programma di sequenziamento 204. Nella forma di attuazione di Figura 1, il programma di sequenziamento risiede sul sistema di computer di realizzazione di fasi 140 mentre nella forma di attuazione di Figura 2, il programma di sequenziamento risiede sul file server 202. Il programma di sequenziamento 204 opera insieme alla base dati 100 (delle Figure 1 e 2) per sequenziare le fasi di installazione del software e di verifica

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

per il sistema di computer obiettivo 160. Quando le fasi di installazione del software e di verifica appropriate per il particolare sistema di computer obiettivo vengono sequenziate, il programma di sequenziamento 204 produce file di uscita come illustrato nel modulo 430.

Nella forma di attuazione illustrata in Figura 1, i file di uscita sono scritti preferibilmente sul disco di fasi 140 (vedere Figura 1) in sei file separati. Questi file comprendono (1) un file di fasi, (2) un file Setenv.bat, (3) un file Qt.txt, (4) un file Et.txt, (5) un Etlast.txt, e (6) un file Ft.txt. E' desiderabile che il file di fasi sia un file di testo ASCII comprendente un elenco di linee di comando appropriate per eseguire le fasi di installazione del software e di verifica per il sistema di computer obiettivo in ordinazione. In una forma di attuazione preferita, il file di fasi comprende inoltre comandi che possono essere eseguiti ad anello. Più specificamente, il file di fasi consente di ripetere i comandi per un numero definito di iterazioni oppure per una durata temporale definita. Tale formato consente in modo vantaggioso di ripetere le fasi di installazione del software o

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

verifica in maniera predeterminata e calcolata. Il file Setenv.bat stabilisce preferibilmente variabili di ambiente sul sistema di computer obiettivo. Si potrà apprezzare che in un modo di funzionamento sono necessari soltanto il file Step e il file Setenv.bat per l'installazione e la verifica. Il file Step e il file Setenv.bat sono file di istruzioni di testo ASCII contenenti un elenco di linee di comando appropriate per eseguire le fasi di installazione e verifica per il sistema di computer obiettivo. Il file Qt.txt, Et.txt, Etlast.txt e Ft.txt sono preferibilmente tutti file di testo ASCII contenenti un elenco di linee di comando appropriate per eseguire le fasi di installazione del software e di verifica per il sistema di computer obiettivo nelle fasi di fabbricazione di Test Rapido (Qt), Test Esteso 1 (Et), Test Esteso 2 (Etlast) e Test Finale (Ft) del sistema di computer obiettivo.

Nella forma di attuazione di Figura 2, d'altra parte, i file di uscita non sono scritti in un disco di fasi come illustrato in Figura 1. Invece, i file di uscita risiedono sul file server 202 o sul file server 190, dove sono utilizzati per dirigere l'esecuzione delle fasi di installazione

software e/o di verifica sul sistema di computer obiettivo 160.

La Figura 5 illustra un diagramma schematico più dettagliato del funzionamento del programma di sequenziamento 204 illustrato nelle Figure 1 e 2. Nel modulo 500, un record di descrittore di sistema corrispondente al sistema di computer obiettivo 160 è fornito al programma di sequenziamento 204. Nel modulo 510, un descrittore di componente viene letto dal record descrittore di sistema. Ogni descrittore di componente descrive un rispettivo componente, hardware o software, nel sistema di computer obiettivo.

Passando alla Figura 3B, la riga del record descrittore di sistema comprendente il processore Pentium[®] nel modulo 370 è un descrittore di componente esemplificativo. Nel modulo 520, il programma di sequenziamento 204 esemplifica una pluralità di oggetti derivati corrispondenti alla pluralità di componenti del sistema di computer obiettivo 160. Nella forma di attuazione preferita, questi oggetti derivati sono utilizzati per memorizzare informazioni (ottenute dalla banca dati 100) relative alle fasi di installazione software e di verifica che occorre eseguire sul sistema di

computer obiettivo 160. Nel modulo 550, le fasi di installazione software e di verifica associate ai rispettivi componenti del sistema di computer obiettivo 160 sono reperite dalla base dati 100 e memorizzate nell'oggetto derivato appropriato. Nella forma di attuazione di Figura 1, le fasi sono reperite tramite la connessione di rete 110 mentre nella forma di attuazione di Figura 2, le fasi possono essere reperite direttamente dal file server 202. Per descrivere come le fasi sono reperite dalla banca dati 100 nella forma di attuazione preferita si richiede una descrizione della costruzione preferita di tale base dati.

La Figura 6 illustra il progetto della base dati 100. La base dati 100 associa sequenze di fasi di installazione del software e/o verifica, in un predeterminato ordine, con famiglie di sistemi di computer. Inoltre, la base dati 100 è configurata in modo da associare i componenti dei sistemi di computer con famiglie di sistemi di computer. Ancor di più, la base dati 100 associa le fasi di installazione del software e/o di verifica con componenti di sistemi di computer.

La base dati 100 è preferibilmente una base dati relazionale. La base dati 100 contiene

parecchie tabelle, contenenti ciascuna attributi adatti a creare le associazioni sopra citate.

La base dati 100 contiene la tabella Step 102, la tabella SysFamily 104, la tabella Sys_Step_Seq 106, la tabella Component 108, la tabella Sys_Comp 112, e la tabella Comp_Step 114. Nella forma di attuazione preferita, ogni tabella contiene un elenco di attributi, gli attributi sottolineati servendo come chiave primaria.

La tabella Step 102 contiene un insieme di fasi di installazione del software e di verifica che sono condivise tra componenti diverse di tutte le famiglie di computer. Nella costruzione preferita, la tabella Step 102 ha attributi comprendenti StepID, Phase, Name, Cmd, CmdType, AfterCode e MaxInstance. StepID è un numero di identificazione unico per ogni fase di installazione del software o di verifica. Phase designa in quale stadio di fabbricazione deve eseguire la fase. Per esempio Phase è un intero scelto in modo da corrispondere a quattro fasi di fabbricazione dei sistemi di computer comprendenti (1) Test Rapido, (2) Test Esteso 1, (3) Test Esteso 2, e (4) Test Finale. Name è una stringa che assegna un nome che è descrittivo della fase. Cmd è

BUZZI, NOTARO &
ANTONIELLI D'ORIO
S.r.l.

una stringa che assegna una linea di comando eseguibile per eseguire la fase di installazione del software o di verifica sul sistema obiettivo 160 (illustrato nelle Figure 1 e 2). AfterCode è un identificatore che determina se è necessaria una interruzione o una nuova inizializzazione dopo che si esegue la fase di installazione del software o di verifica. MaxInstance è un identificatore che indica il numero massimo di volte consentite per cui può essere eseguita la fase. Infine, ClassID identifica un certo tipo di componente che è associato alla fase di installazione del software o di verifica.

La tabella SysFamily 104 identifica ogni famiglia di sistemi di computer con un intero di identificazione specificato nell'attributo SysID. E' compresa anche nella tabella SysFamily una stringa che identifica il nome della famiglia.

La tabella Sys_Step_Seq 106 è una tabella relazionale che contiene le relazioni tra la tabella Step 102 e la tabella SysFamily 104. La tabella Sys_Step_Seq 106 comprende un intero di identificazione di famiglia specificato nell'attributo SysID per una particolare famiglia di sistemi di computer (dalla tabella SysFamily

104), un intero di identificazione di fase specificato nell'attributo StepID (dalla tabella Step 102) che identifica un particolare insieme di fasi appropriate per tale famiglia, ed un numero di sequenza. Il numero di sequenza è contenuto preferibilmente all'interno dell'attributo SeqNum che rappresenta un predeterminato ordine in cui devono essere eseguite le fasi associate ad una particolare famiglia. Gli ingegneri di test assegnano i numeri di sequenza, unici nell'ambito di ogni stadio di fabbricazione, in un ordine scelto in modo che sia più efficace per un particolare sistema obiettivo. Si potrà apprezzare che si possono utilizzare altri modi per assegnare i numeri di sequenza.

La tabella Component contiene tutti i possibili componenti che sono inseriti all'interno dei sistemi di computer in fabbricazione. Gli attributi di questa tabella sono preferibilmente CompID che assegna un identificatore ad ogni componente, NameDesc che assegna un nome di stringa ad ogni componente, e CallsId che fa riferimento al tipo di componente (ad esempio unità a disco rigido, unità CD-ROM).

La tabella Sys_Comp 112 è una tabella

relazionale contenente relazioni tra una famiglia di sistemi di computer ed un insieme di componenti che possono essere inseriti in tale famiglia. Gli attributi della tabella Sys_Comp 112 comprendono un intero di identificazione di famiglia di computer specificato nell'attributo SysID (dalla tabella SysFamily 104) ed un intero di identificazione di componente specificato nell'attributo CompID (dalla tabella Component 108).

La tabella Comp_Step 114 è una tabella relazionale contenente le relazioni tra un componente ed un insieme di fasi di installazione software e di verifica appropriate per tale componente. Gli attributi della tabella Comp_Step 114 comprendono un intero di identificazione di componente specificato nell'attributo CompID (dalla tabella Component 108) ed un intero di identificazione di fase specificato nell'attributo StepID (dalla tabella Step 102).

Il sistema di computer obiettivo esemplificativo illustrato in Figura 3B sarà utilizzato per illustrare come si utilizza il progetto di base dati sopra evidenziato per reperire le fasi di installazione del software e di verifica. L'identificatore di famiglia di computer

nel record descrittore di sistema che identifica la famiglia X è associato al SysID corrispondente alla famiglia X nella tabella SysFamily 104. Si utilizza la tabella Component 108 per controllare se i componenti del sistema di computer obiettivo elencati nell'ordine di sistema di computer obiettivo sono legali. In altre parole, il programma di sequenziamento e la base dati determinano se il processore, l'unità a disco rigido, il monitor e il software contenuti nel record descrittore di sistema di Figura 3B hanno corrispondenti inserimenti e corrispondenti interi specificati da CompID nella tabella Component 108. Se un componente non è legale (cioè se un componente nel record descrittore di sistema non è contenuto nella tabella Component 108), si crea un indicatore di errore. La tabella Sys_Comp 112 è una tabella relazionale che contiene le mappature dalla tabella Component 108 e dalla tabella SysFamily 104. La tabella Sys_Comp 112 contiene tutti i componenti legali che possono essere compresi su un sistema di computer obiettivo appartenente alla famiglia X. Così, si può utilizzare la tabella Sys_Comp 112 per controllare se tutti i componenti del sistema obiettivo sono legali. In altre parole,

SUZZI, NOTARO &
ANTONIELLO D'OUX
S.r.l.

il programma di sequenziamento e la base dati determinano se il processore, l'unità a disco rigido, il monitor e il software contenuti nel record descrittore di sistema di Figura 3B hanno corrispondenti relazioni nella tabella Sys_Comp 112. Se un componente non è legale (cioè se un componente nel record descrittore di sistema non può essere compreso su un sistema obiettivo appartenente alla famiglia X), si crea un indicatore di errore.

Nella tabella relazionale Sys_Step_Seq 106 risiedono le mappature dalla tabella Step 102 e dalla tabella Sys_Family 104. La tabella Sys_Step_Seq 106 contiene tutte le fasi di installazione del software e di verifica che possono essere fatte funzionare legalmente sui sistemi di computer obiettivo appartenenti alla famiglia X. Inoltre, è in questa tabella Sys_Step_Seq 106 che i numeri di sequenza e di fase sono associati ad ogni fase di installazione del software e di verifica. Questi numeri di sequenza e fase rappresentano l'ordine appropriato in cui le fasi devono essere eseguite per una particolare famiglia di sistemi di computer. Pertanto, la tabella Sys_Step_Seq 106 contiene un elenco di fasi

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

che devono essere eseguite sui sistemi di computer obiettivo della famiglia X così come i numeri di sequenza di fase che rappresentano un predeterminato ordine in cui le fasi devono essere eseguite.

La tabella Comp_Step 114 è una tabella relazionale che contiene le mappature dalla tabella Component 108 e dalla tabella Step 102. La tabella Comp_Step 114 contiene le fasi di installazione del software e di test che devono essere eseguite per il processore, l'unità a disco rigido, il monitor e il software del sistema di computer obiettivo.

Per reperire le fasi di installazione del software e di verifica associate ai rispettivi componenti da inserire sul sistema obiettivo si richiede l'esecuzione di una operazione di unione sulla tabella Sys_Comp 112 e sulla tabella Comp_Step 114 per ottenere un insieme intermedio che elenca le fasi che devono essere eseguite sui componenti del sistema di computer obiettivo 160.

L'operazione di unione dà come risultato un elenco di fasi che devono essere eseguite sul processore, sull'unità a disco rigido, sul monitor e sul software elencati nel record descrittore di sistema illustrato in Figura 3B. Il risultato

dell'unione della tabella Sys_Comp 112 e della tabella Comp_Step 114 viene quindi unito alla tabella Sys_Step_Seq 106 che contiene tutte le fasi per la famiglia X. Il risultato di questa operazione di unione comprende le informazioni di sequenziamento sotto forma di numeri di sequenza e numeri di fase, i numeri di sequenza essendo unici all'interno di una fase particolare. Così, una unione a tre tabelle della tabella Sys_Comp 112, della tabella Comp_Step 114 e della tabella Sys_Step_Seq 106 produce le fasi di installazione del software e di verifica appropriate così come le informazioni di sequenziamento sotto forma di numeri di sequenza e fase per installare e/o testare il software sul sistema di computer obiettivo 160.

Se il risultato della prima operazione di unione (l'unione della tabella Sys_Comp 112 e della tabella Comp_Step 114) è un insieme vuoto, occorre creare una condizione di errore, dato che un insieme vuoto segnala che un componente da inserire sul sistema obiettivo non appartiene alla famiglia elencata sul record descrittore di sistema. Un esempio di ciò è illustrativo. Si consideri che un record di descrittore di sistema indica

correttamente che un sistema di computer obiettivo appartiene alla famiglia X. Si ipotizzi, tuttavia, che il record descrittore di sistema indichi non correttamente che un'unità a disco rigido (unità a disco rigido Z) appartenente soltanto a sistemi obiettivo nella famiglia X debba essere inserita sul sistema obiettivo che è nella famiglia Y. In tal caso, la tabella Comp_Step 114 contiene le fasi associate all'unità a disco rigido Z. La tabella Sys_Comp 112 contiene i componenti associati alla famiglia Y. Così, l'unione della tabella Comp_Step 114 con la tabella Sys_Comp 112 produce un insieme vuoto, dato che l'unità a disco rigido Z non è un componente associato alla famiglia Y (invece, esso è associato soltanto alla famiglia X). Come appare chiaro dal suddetto esempio, il progetto preferito della base dati consente in modo vantaggioso di accertarsi che un sistema obiettivo di una certa famiglia contenga soltanto i componenti appropriati per tale famiglia.

Con riferimento di nuovo alla Figura 5, dopo che si reperiscono le fasi associate ai componenti da inserire nel sistema obiettivo, il programma di sequenziamento 204 prepara le variabili di ambiente per il sistema di computer obiettivo nel modulo 560

BUZZI, NOTARO &
ANTONIELLI D'OUX
S.r.l.

leggendo il record descrittore di sistema e creando un file di ambiente corrispondente ai componenti da inserire sul sistema obiettivo. Per esempio, si legge il record descrittore di sistema illustrato in Figura 3B, e si potrebbe preparare una variabile di ambiente quale "set cpu=pentium" corrispondente al componente hardware di processore del record descrittore di sistema.

Nel modulo 570 di Figura 5, la pluralità di fasi di installazione del software e di verifica, reperite tramite l'unione di tre tabelle sopra descritte, sono sequenziate nell'ordine predeterminato. Questo sequenziamento avviene secondo i rispettivi numeri di sequenza e numeri di fase per fornire una sequenza di fasi. Il sequenziamento stesso deve essere realizzato utilizzando uno qualsiasi tra molti algoritmi di ordinamento ben noti nella tecnica.

Nel modulo 580, il programma di sequenziamento 204 invia in uscita i file. Come citato in precedenza, i file di uscita sono scritti preferibilmente sul disco di fasi 150 (vedere Figura 1) in sei file separati nella forma di attuazione illustrata in Figura 1. Questi file comprendono (1) un file Step, (2) un file

Setenv.bat, (3) un file Qt.txt (4), un file Et.txt (5) un file Etlast.txt e (6) un file Ft.txt. E' desiderabile che il file Step sia un file di testo ASCII. Nella forma di attuazione preferita, il file Step comprende inoltre comandi che possono essere eseguiti ad anello. Più specificamente, il file Step consente di ripetere i comandi per un numero definito di iterazioni o per una durata di tempo definita. Il file Setenv.bat stabilisce le variabili di ambiente sul sistema di computer obiettivo. Il file di fasi contiene le fasi che devono essere eseguite rispettivamente durante le stadi di fabbricazione di Test Rapido (Qt), Test Esteso 1 (Et), Test Esteso 2 (Etlast), e Test Finale (Ft) del sistema di computer obiettivo. Nella forma di attuazione di Figura 2, d'altra parte, i file di uscita non sono scritti in un disco di fasi come illustrato in Figura 1. Invece, i file di uscita risiedono sul file server 202 o sul file server 190, dove possono essere utilizzati per dirigere l'esecuzione delle fasi di installazione del software e di verifica sul sistema di computer obiettivo 160.

Passando ora alle Figure 1 e 2, la freccia 130 illustra il fatto che si possono effettuare

modifiche sulla base dati 100. Per esempio, se si crea una nuova famiglia di sistemi di computer, si può modificare la base dati 100 di conseguenza. Più specificamente, alla nuova famiglia viene assegnato un identificatore di famiglia nuova in SysID della tabella SysFamily 104 e si assegna un nome per la famiglia nell'attributo Name della tabella Sys_Family 104. Si aggiungono un elenco di fasi di installazione di software e di fasi di verifica alla tabella Sys_Step_Seq 106, queste fasi rappresentando quali fasi devono essere eseguite, e in quale ordine predeterminato, sulla nuova famiglia di sistemi di computer. Se la nuova famiglia di sistemi di computer condivide parecchie somiglianze con una famiglia esistente, è probabile che gli inserimenti per la famiglia esistente nella tabella Sys_Step_Seq 106 possano essere modificati per produrre inserimenti per la nuova famiglia. Se occorre creare qualsiasi fase nuova per la nuova famiglia di sistemi di computer, queste fasi sono aggiunte alla tabella Step 102. Analogamente, se qualche nuovo componente accompagna la nuova famiglia di sistemi di computer, questi componenti vengono aggiunti alla tabella Component 108. La tabella Comp_Step 114 viene aggiornata per

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

associare ogni componente della nuova famiglia di sistemi di computer nelle fasi appropriate per la sua installazione del software e verifica. Se la nuova famiglia utilizza soltanto componenti già presenti nella base dati, non occorre modificare questa tabella. La tabella Sys_Comp 112 viene aggiornata in modo che un elenco di componenti consentiti che possono essere compresi nella nuova famiglia sia nella base dati. In particolare, sarebbe necessario associare il SysID del nuovo sistema di computer con il CompID di ogni componente consentito. Di nuovo, si potrebbe fare ciò copiando e quindi modificando un inserimento esistente di una famiglia più vecchia di sistemi di computer.

Si potrà apprezzare che nella costruzione di una base dati secondo la forma di attuazione preferita, sono previsti certi vantaggi significativi. In particolare, il progetto modulare della base dati consente in modo vantaggioso una facile impostazione delle fasi di installazione del software e di verifica per nuove famiglie di sistemi di computer. In aggiunta, le fasi di installazione del software e verifica per una particolare famiglia di sistemi di computer o per

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

un particolare componente possono essere modificate indipendentemente da altre fasi di installazione del software e di verifica.

La Figura 7 illustra come un record descrittore di sistema ed una sequenza di fasi possono essere sottoposti a patch per consentire modifiche modulari in un processo di installazione del software e di verifica in conformità con l'invenzione. Nel modulo 600, si crea un record descrittore di sistema. Nel modulo 610, il record descrittore di sistema viene modificato utilizzando un patch di record descrittore di sistema. Nella forma di attuazione preferita, questa patch è modulare, consentendo di creare patch per un sistema di computer obiettivo specifico, una particolare famiglia di sistemi di computer, o per un particolare componente. Per esempio, se un costruttore volesse sostituire una marca di unità a disco rigido con un'altra per una certa famiglia di sistemi di computer in un certo giorno, si potrebbe formare un patch che modificherebbe tutti i record descrittore di sistema contenenti l'unità a disco rigido da sostituire ed effettuare la sostituzione nel modulo 610. Nel modulo 620, si determina una sequenza di fasi come evidenziato in precedenza.

Nel modulo 630, si modifica la sequenza di fasi utilizzando un patch di sequenza di fasi. Nella forma di attuazione preferita, questo patch è modulare, consentendo di creare patch per un sistema di computer obiettivo specifico, una particolare famiglia di sistemi di computer o per un particolare componente. Per esempio, se un costruttore volesse eseguire una fase di verifica prima di un'altra per un certo componente un certo giorno, si potrebbe formare un patch che modificherebbe tutte le sequenze di fasi contenenti le fasi il cui ordine deve essere modificato e modificherebbe in modo corrispondente l'ordine di esecuzione nel modulo 640.

Si presterà ora l'attenzione all'esecuzione della sequenza di fasi sul sistema obiettivo 160. Le fasi di installazione del software o di verifica vengono eseguite sul sistema di computer obiettivo 160 utilizzando un programma che legge, interpreta ed esegue la sequenza di fasi corrispondente al sistema di computer obiettivo. Nella forma di attuazione preferita, questo programma è chiamato Runstep ed è posto sul disco di fasi 150 nella forma di attuazione di Figura 1 e sul file server 202 nella forma di attuazione di Figura 2.

La Figura 8 illustra una porzione di una sequenza di fasi contenuta in un file di fasi prima che si esegua qualsiasi fase di installazione del software e di verifica. Come citato in precedenza, la sequenza di fasi comprende comandi per installare il software e/o per verificare il sistema di computer obiettivo prodotto su ordinazione. In aggiunta, la sequenza di fasi nelle file di fasi consente di ripetere i comandi per un numero definito di iterazioni o per un periodo di tempo definito. Inoltre, il file di fasi può contenere commenti, ignorati dal programma Runstep. Nel file di fasi, si utilizzano segni 800 per separare i campi della sequenza di fasi. L'elemento 810 sono comandi per verificare il sistema di computer obiettivo 160. I comandi comprendono, per esempio, un comando per verificare la memoria e per verificare i dispositivi di interfaccia di sistema di computer piccola (SCSI). Come si può vedere dalla figura, ogni comando può comprendere interruttori quali '-o' appropriati per il particolare ambiente di verifica. L'elemento 820 è un commento che è ignorato dal programma Runstep. L'elemento 810c è un comando che viene eseguito ad anello nel tempo. Nella costruzione preferita,

BUZZI, NOTARO &
ANTONIELLI D'OUIX
s.r.l.

l'istruzione 'begin_time_loop' indica il punto iniziale di un anello. L'istruzione 'end_time_loop' indica il punto terminale di un anello. L'istruzione 'begin_time_loop' si combina con un campo che indica la durata temporale per le iterazioni attraverso l'anello. Qui, per esempio, il comando 810c viene eseguito per un'ora e trenta minuti. L'elemento 810d è un comando che viene eseguito ad anello secondo un certo numero di iterazioni. Nella forma di attuazione preferita, il comando 'begin_iterate_loop' istruisce il programma Runstep che occorre eseguire un anello iterativo. Il comando 'end_iterate_loop' segnala la fine dei comandi di esecuzione ad anello. Qui, il comando 810d è eseguito tre volte.

Quando il programma Runstep esegue la sequenza di fasi, il programma Runstep colloca informazioni di indicatori temporali nel file di fasi, consentendo in modo vantaggioso una facile risoluzione degli errori e un rintracciamento del processo di installazione del software e di verifica.

La Figura 9 illustra una porzione della sequenza di fasi di Figura 8 dopo che le fasi vengono eseguite. Come illustrato, il programma

Runstep inserisce informazioni di indicazioni temporali nella sequenza di fasi. L'elemento 830 illustra quando è cominciato il test di memoria, e l'elemento 832 illustra quando tale test è terminato. L'elemento 834 illustra quando ha cominciato l'ultima iterazione del test. L'elemento 836 e 838 illustrano quando il test scsiHD è cominciato ed è terminato, rispettivamente. L'elemento 840 conferma che l'anello iterativo è stato eseguito tre volte. Infine, gli elementi 842 e 844 illustrano quando l'ultima iterazione del test scsiCD è cominciata ed è terminata, rispettivamente. L'inserimento di informazioni di indicazioni temporali adiacenti al comando che era stato eseguito consente in modo vantaggioso una efficiente risoluzione degli errori e un rintracciamento del processo di installazione del software e di verifica.

La Figura 10 illustra il flusso generale preferito del programma Runstep. Il programma Runstep 860 è eseguito in un anello con un file batch Runstep 870. Il programma Runstep 860 legge e interpreta una fase in una sequenza di fasi e scrive il comando che deve essere eseguito dalla sequenza di fasi nel file batch 870. Il file batch

870 viene quindi eseguito, eseguendo la fase sul sistema di computer obiettivo 160. Al completamento di una fase, il controllo ritorna dal file batch al programma Runstep 860 che quindi legge e interpreta la riga successiva della sequenza di fasi.

La Figura 11 illustra un flusso più dettagliato del programma Runstep. Come illustrato nel modulo 900, il programma Runstep controlla dapprima per vedere se esiste un file chiamato Re_Run.bat. Un file Re_Run.bat viene creato prima che un qualsiasi comando sia eseguito da una sequenza di fasi e viene tolto dopo il completamento positivo del comando. L'esistenza di Re_Run.bat indica al programma Runstep nel modulo 900 che l'ultima esecuzione del comando non è stata completata positivamente. Così Re_Run.bat funziona come indicazione di inizio dell'esecuzione. Se Re_Run.bat esiste, si chiede ad un operatore nel modulo 904 se il processo di installazione del software e di verifica debba continuare o meno oppure se l'operatore preferisce invece eseguire la ricerca e risoluzione degli errori. Se un operatore sceglie di continuare, il controllo passa al modulo di esecuzione 928 dove si riesegue il file Runstep.bat. (Questa condizione è la scelta di

BUZZI, NOTARO &
ANTONIELLI D'OUDEX
s.r.l.

default se non si sceglie affermativamente alcuna opzione). Se si sceglie l'opzione di ricerca e risoluzione di errori, si esegue la risoluzione degli errori come è ben noto nella tecnica.

Se Re_Run.bat non esiste, il programma Runstep determina che l'ultimo comando è stato completato correttamente, e il controllo viene passato al modulo 910, dove si legge una riga della sequenza di fasi, contenuta preferibilmente in file di fasi. Il programma Runstep legge la linea e determina se esiste un indicatore temporale di inizio e di fine del modulo 912. Se esiste un indicatore di inizio o di fine, il programma Runstep determina, nel modulo 914, se esiste soltanto un indicatore temporale di inizio per la riga che il programma Runstep sta leggendo. Se esiste soltanto un indicatore temporale di inizio, il programma Runstep ipotizza nel modulo 916 che una fase di installazione del software o di verifica sia appena finita e riempie un indicatore temporale di fine nel modulo 918. Dopo aver riempito l'indicatore temporale di fine, il controllo ritorna al modulo 900.

Se esiste più che un solo indicatore temporale di inizio per la riga che il programma Runstep sta leggendo, il programma Runstep determina nel modulo

906 se esistono sia un indicatore temporale di inizio sia uno di fine. Se è così, il programma Runstep ipotizza nel modulo 908 che la fase sia stata eseguita e il controllo ritorna al modulo 900. Se il programma Runstep non incontra alcun indicatore temporale di inizio o di fine nel modulo 912, il programma Runstep riempie l'indicatore temporale di inizio nel modulo 920 e si prepara a eseguire la fase sulla riga della sequenza di fasi che il programma Runstep sta leggendo.

Nel modulo 922, il programma Runstep determina se il comando da eseguire è memorizzato su una unità locale (il file di fasi controlla quale unità nel sistema è l'unità locale). L'unità locale può essere, ad esempio, il disco di fasi, un'unità a disco rigido del sistema obiettivo, un'unità RAM del sistema obiettivo, oppure un'unità di rete. Se il comando non è posto sull'unità locale, il programma Runstep ipotizza che il test da eseguire sia contenuto su un file server da qualche parte su una rete. Il programma Runstep determina nel modulo 932 se il programma Runstep è già connesso a tale rete. Se non così, il programma Runstep, nel modulo 936, incorpora un comando in Runstep.bat per il collegamento alla rete. Pertanto si effettua una

BUZZI, NOTARO &
ANTONELLI D'OUDE
s. r. l.

connessione di rete prima che Runstep.bat esegua la fase sul sistema di computer obiettivo 160 sulla connessione di rete 180.

A seguito del modulo 936, il controllo viene passato al modulo 926. Se il programma Runstep è già connesso nella rete, il programma Runstep, durante il modulo 934, toglie i comandi da Runstep.bat per il collegamento alla rete, dato che una fase di collegamento aggiuntiva non è necessaria se esiste già una connessione di rete. Il controllo viene quindi passato al modulo 926. Se succede che la fase da eseguire si trova sul disco di fase 150, non occorre che il programma Runstep si colleghi alla rete. Così, nel modulo 924, il programma Runstep toglie i comandi da Runstep.bat per il collegamento alla rete. Il controllo viene quindi passato al modulo 926. Nel modulo 926, il programma Runstep incorpora il comando appropriato da eseguire in Runstep.bat e in Re_run.bat. Il comando così incorporato viene preso dalla sequenza di fasi, contenuta preferibilmente nel file di fasi. Nel modulo 928, la fase viene eseguita facendo girare Runstep.bat e, se eseguita positivamente, si cancella Re_run.bat. Se la fase non viene eseguita positivamente, il file

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

Re_Run.bat non viene cancellato e il controllo si trasferisce allo stato di errore 929. Il controllo viene quindi restituito al modulo 900 in modo che si possa leggere un'altra riga dalla sequenza di fasi. Questo processo continua fino a quando tutte le fasi di installazione del software e di verifica sono completate.

All'esecuzione della sequenza di fasi, il sistema obiettivo viene verificato e si installa il software. Nella forma di attuazione di Figura 1, si possono eseguire un numero selezionato di test direttamente dal disco di fasi 150, ma la maggior parte dei test sono eseguiti dal file server 190 sulla connessione di rete 180. L'esecuzione dei test dal file server 190 elimina in modo vantaggioso i limiti imposti dalle capacità di memorizzazione dei dischi floppy quali il disco di fasi 150.

Nella forma di attuazione di Figura 2, le fasi sono eseguite dal file server 190 sulla connessione di rete 180. Un disco floppy, qui il disco di boot 220, è necessario soltanto per inizializzare il sistema di computer obiettivo 160. Tale sistema semplifica in modo vantaggioso il processo di installazione del software e di verifica.

Passando ancora una volta alle Figure 1 e 2, la freccia 210 illustra il fatto che i risultati dalla installazione del software e dalle verifiche possono essere registrati nuovamente sul file server 190 o sul file server 202. I risultati comprendono preferibilmente il fatto che tutte le fasi sono state completate positivamente e quali tipi di errori (se esistono) si sono incontrati. La registrazione dei risultati potrebbe comprendere il semplice salvataggio o scrittura di una versione modificata del file di fasi a seguito dell'esecuzione della sequenza di fasi, dato che come discusso in precedenza il file di fasi ha una indicazione temporale tramite il programma di Runstep. Tale sistema consente in modo vantaggioso migliori capacità di risoluzione degli errori durante la fabbricazione del sistema di computer.

Mentre forme di attuazione particolari della presente invenzione sono state illustrate e descritte, risulterà ovvio agli esperti nel ramo che si possono effettuare cambiamenti e modifiche senza discostarsi dalla presente invenzione nei suoi aspetti più ampi e, pertanto, le rivendicazioni allegate servono a comprendere

all'interno del loro campo di protezione tutti questi cambiamenti e modifiche come ricadono all'interno del vero spirito e campo di protezione della presente invenzione.

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

RIVENDICAZIONI

1. Procedimento di installazione del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

 accedere ad un comando all'interno della sequenza di fasi;
 creare un file comprendente un'indicazione dell'inizio di esecuzione;
 eseguire un comando; e
 cancellare il file comprendente l'indicazione dell'inizio di esecuzione quando il comando completa l'esecuzione; e
 ripetere le fasi di accedere, creare, eseguire e cancellare per la pluralità di fasi.

2. Procedimento secondo la rivendicazione 1, comprendente inoltre le fasi di:

 determinare se il file comprendente

l'indicazione dell'inizio di esecuzione è presente prima di accedere ad un comando all'interno della sequenza di fasi; e determinare che è presente una condizione di errore se il file comprendente l'indicazione dell'inizio di esecuzione è presente.

3. Procedimento secondo la rivendicazione 2, comprendente inoltre le fasi di:

chiedere ad un operatore se l'operatore desidera continuare con l'installazione del software quando è presente la condizione di errore; e

eseguire il comando se l'operatore desidera continuare con l'installazione del software.

4. Procedimento secondo la rivendicazione 2 comprendente inoltre le fasi di:

chiedere ad un operatore se l'operatore desidera trasferirsi in un modo di funzionamento di risoluzione degli errori quando è presente la condizione di errore; e

trasferirsi ad un modo di funzionamento di correzione degli errori se l'operatore lo

desidera.

5. Procedimento per l'installazione del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

leggere un comando dalla sequenza di fasi;

ricercare il comando in una unità locale del sistema di computer;

eseguire il comando dall'unità locale se il comando è memorizzato sull'unità locale del sistema di computer;

eseguire il comando da un server se il comando non è memorizzato sull'unità locale del sistema di computer.

6. Procedimento secondo la rivendicazione 5, comprendente inoltre le fasi di:

determinare se il sistema di computer è collegato su una rete; e

collegarsi alla rete se il sistema di computer non è collegato sulla rete.

7. Procedimento secondo la rivendicazione 6 comprendente inoltre le fasi di:

incorporare istruzioni di collegamento in un file batch per collegarsi ad una rete se il comando è ubicato su un file server connesso alla rete in comunicazione con il sistema di computer;

incorporare il comando nel file batch;

eseguire il file batch per eseguire il comando sul sistema di computer.

8. Procedimento secondo la rivendicazione 6, in cui una fase all'interno del file di fasi è un comando di collegamento alla rete e comprendente inoltre le fasi di:

togliere il comando di collegamento alla rete dalla sequenza di fasi se il sistema di computer è collegato sulla rete.

9. Procedimento per l'installazione del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase

BUZZI, NOTARO &
ANTONELLI D'OUX
s.r.l.

essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

leggere un primo comando dalla sequenza di fasi;

incorporare istruzioni di collegamento in un primo ed un secondo file batch per il collegamento ad una rete se il primo comando è ubicato su un file server connesso alla rete in comunicazione con il sistema di computer;

togliere le istruzioni di connessione alla rete dal primo e dal secondo file batch se il primo comando è ubicato su una unità locale del sistema obiettivo;

incorporare il primo comando nel primo e nel secondo file batch;

eseguire il primo file batch per eseguire il primo comando sul sistema di computer;

togliere il secondo file batch al completamento positivo del primo comando;

leggere un secondo comando dalla sequenza di fasi se il secondo file batch è stato

BUZZI, NOTARO &
ANTONIELLI D'OUX
s.r.l.

tolto.

10. Procedimento per l'installazione del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

- leggere un comando dalla sequenza di fasi;
- eseguire il comando sul sistema di computer;
- creare un indicatore temporale di inizio che indica quando è cominciata l'esecuzione del comando; e
- creare un indicatore temporale di fine che indica quando è terminata l'esecuzione del comando.

11. Procedimento secondo la rivendicazione 10, in cui il comando è eseguito ripetutamente per un periodo di tempo definito.

12. Procedimento secondo la rivendicazione 11, in cui gli indicatori temporali di inizio e di fine

indicano quando l'esecuzione di una iterazione finale del comando è cominciata e terminata, rispettivamente.

13. Procedimento secondo la rivendicazione 10, in cui il comando è eseguito ripetutamente per un numero di volte definito.

14. Procedimento secondo la rivendicazione 13, in cui gli indicatori temporali di inizio e di fine indicano quando l'esecuzione di una iterazione finale del comando è cominciata e terminata, rispettivamente.

15. Procedimento secondo la rivendicazione 10, in cui gli indicatori temporali di inizio e di fine sono scritti adiacenti al comando nella sequenza di fasi.

16. Procedimento secondo la rivendicazione 10, in cui il comando è collocato su un disco floppy del sistema di computer.

17. Procedimento secondo la rivendicazione 10, in cui il comando è collocato su un file server in comunicazione con il sistema di computer.

18. Procedimento per la verifica del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi

comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componenti, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

 accedere ad un comando all'interno della sequenza di fasi:

 creare un file comprendente una indicazione dell'inizio di esecuzione;

 eseguire il comando; e

 cancellare il file comprendente l'esecuzione dell'inizio di esecuzione quando il comando completa l'esecuzione; e

 ripetere le fasi di accedere, creare, eseguire e cancellare per la pluralità di fasi.

19. Procedimento secondo la rivendicazione 18, comprendente inoltre le fasi di:

 determinare se il file comprendente l'indicazione dell'inizio di esecuzione è presente prima di accedere ad un comando all'interno della sequenza di fasi; e

 determinare che è presente una condizione di errore se il file comprendente

l'indicazione di inizio di esecuzione è presente.

20. Procedimento secondo la rivendicazione 19, comprendente inoltre le fasi di:

chiedere ad un operatore se l'operatore desidera continuare con la verifica del software quando è presente la condizione di errore; e

eseguire il comando se l'operatore desidera continuare con la verifica del software.

21. Procedimento secondo la rivendicazione 19, comprendente inoltre le fasi di:

chiedere ad un operatore se l'operatore desidera trasferirsi ad un modo di funzionamento o di risoluzione degli errori quando è presente la condizione di errore; e

trasferirsi ad un modo di funzionamento di correzione degli errori se l'operatore lo desidera.

22. Procedimento di verifica del software per un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase

comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

- leggere un comando dalla sequenza di fasi;
- ricercare il comando in una unità locale del sistema di computer;
- eseguire il comando dall'unità locale se il comando è memorizzato sull'unità locale del sistema di computer;
- eseguire il comando da un server se il comando non è memorizzato sull'unità locale del sistema di computer.

23. Procedimento secondo la rivendicazione 22, comprendente inoltre le fasi di:

- determinare se il sistema di computer è collegato su una rete; e
- collegarsi alla rete se il sistema di computer non è collegato sulla rete.

24. Procedimento secondo la rivendicazione 23 comprendente inoltre le fasi di: incorporare le istruzioni di collegamento in un file batch per collegarsi ad una rete se il comando è ubicato su

un file server connesso alla rete in comunicazione con il sistema di computer;

incorporare il comando nel file batch;

eseguire il file batch per eseguire il comando sul sistema di computer.

25. Procedimento secondo la rivendicazione 23, in cui una fase all'interno del file di fasi è un comando di collegamento alla rete; e comprendente inoltre le fasi di:

togliere il comando di collegamento alla rete dalla sequenza di fasi se il sistema di computer è collegato sulla rete.

26. Procedimento per la verifica del software di un sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

leggere un primo comando dalla sequenza di fasi;

incorporare istruzioni di collegamento in un primo e un secondo file batch per il collegamento ad una rete se il primo comando è ubicato su un file server connesso alla rete in comunicazione con il sistema di computer;

togliere le istruzioni di collegamento alla rete dal primo e dal secondo file batch se il primo comando è posto su una unità locale del sistema obiettivo;

incorporare il primo comando nel primo e nel secondo file batch;

eseguire il primo file batch per eseguire il primo comando sul sistema di computer;

togliere il secondo file batch al completamento positivo del primo comando;

leggere un secondo comando dalla sequenza di fasi se il secondo file batch è stato tolto.

27. Procedimento per la verifica del software per il sistema di computer avente una pluralità di componenti, il sistema di computer avendo una sequenza di fasi associata, la sequenza di fasi comprendendo una pluralità di fasi, ogni fase comprendendo almeno un comando, una fase essendo

associata ad un rispettivo descrittore di componente, un descrittore di componente corrispondendo ad un rispettivo componente del sistema di computer, il procedimento comprendendo le fasi di:

- leggere un comando dalla sequenza di fasi;
- eseguire il comando sul sistema di computer;
- creare un indicatore temporale di inizio che indica quando è cominciata l'esecuzione del comando; e
- creare un indicatore temporale di fine che indica quando l'esecuzione del comando è terminata.

28. Procedimento secondo la rivendicazione 27, in cui il comando è eseguito ripetutamente per un periodo di tempo definito.

29. Procedimento secondo la rivendicazione 28, in cui gli indicatori temporanei di inizio e di fine indicano quando l'esecuzione di una iterazione finale di comando è cominciata ed è finita, rispettivamente.

30. Procedimento secondo la rivendicazione 27, in cui il comando è eseguito ripetutamente per un numero di volte definito.

31. Procedimento secondo la rivendicazione 30,

in cui gli indicatori temporali di inizio e di fine indicano quando l'esecuzione di una iterazione finale del comando è cominciata ed è finita, rispettivamente.

32. Procedimento secondo la rivendicazione 27, in cui gli indicatori temporali di inizio e di fine sono scritti adiacenti al comando nella sequenza di fasi.

33. Procedimento secondo la rivendicazione 27, in cui il comando è posto su una unità locale del sistema di computer.

34. Procedimento secondo la rivendicazione 27, in cui il comando è posto su un file server in comunicazione con il sistema di computer.

35. Sistema di computer comprendente:

un processore;

un componente accoppiato al processore; e

una memoria accoppiata al processore, la memoria comprendendo il software installato su di essa, il software essendo installato tramite le fasi di:

accedere ad un comando di installazione del

software all'interno di una sequenza di

fasi avente una pluralità di fasi;

creare un file comprendente una indicazione

dell'inizio di esecuzione;
eseguire il comando; e
cancellare il file comprendente l'indicazione
dell'inizio di esecuzione quando il comando
completa l'esecuzione; e
ripetere le fasi di accedere, creare, eseguire
e cancellare per la pluralità di fasi.

19. - Procedimento e sistema sostanzialmente come
descritto ed illustrato e per gli scopi
specificati

Ing. Franco BUZZI

N. iscriz. ALBO 259

Ma proprio e per gli altri



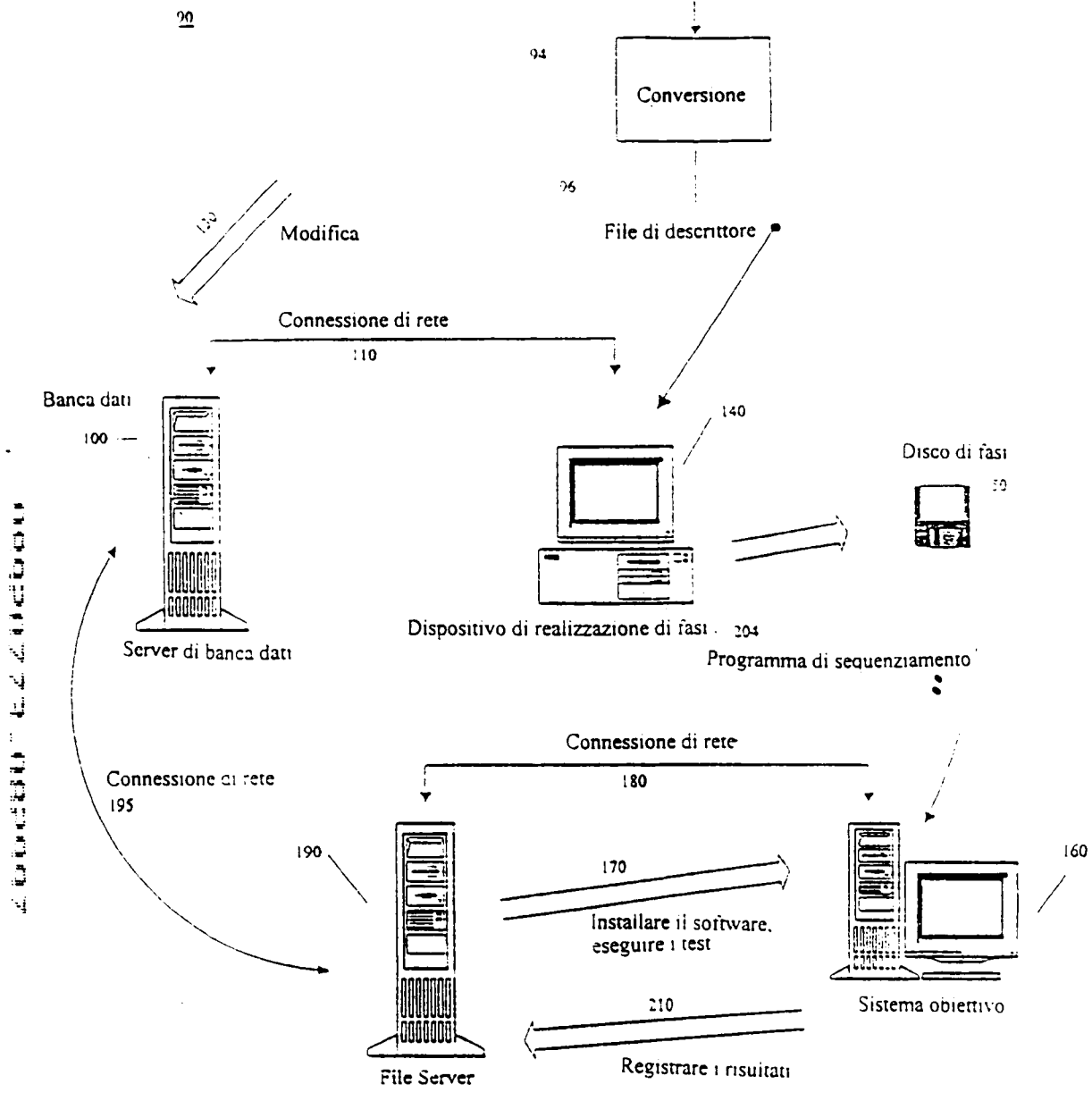


Figura 1



Ing. Franco BUZZI
 N. Iscritt. ALBO 259
 (a proprio e per gli altri)

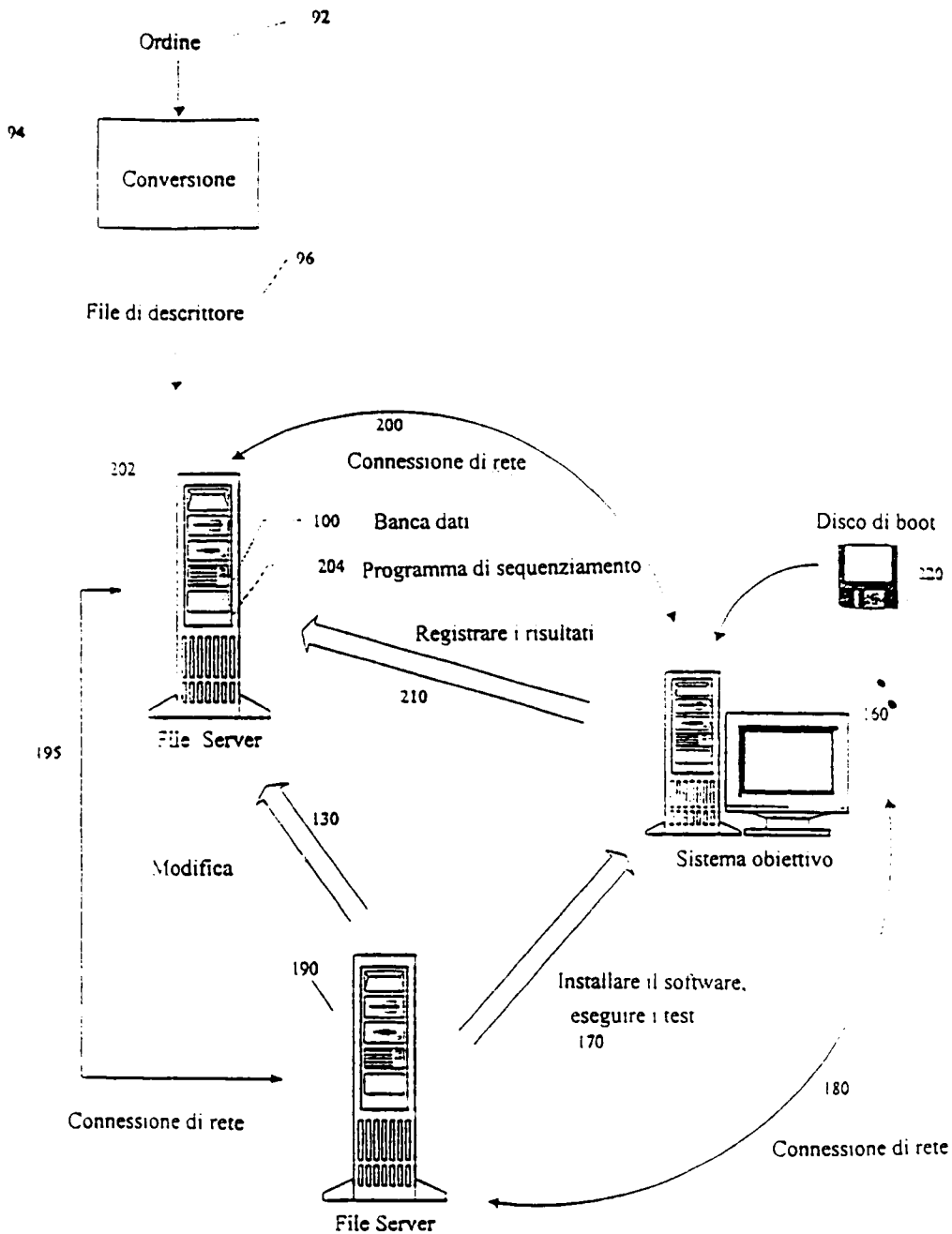


Figura 2



Ing. Franco BUZZI
N. iscriz. ALBO 258
(Io proprio e per gli altri)

0992073-082997
666280-EZ02660

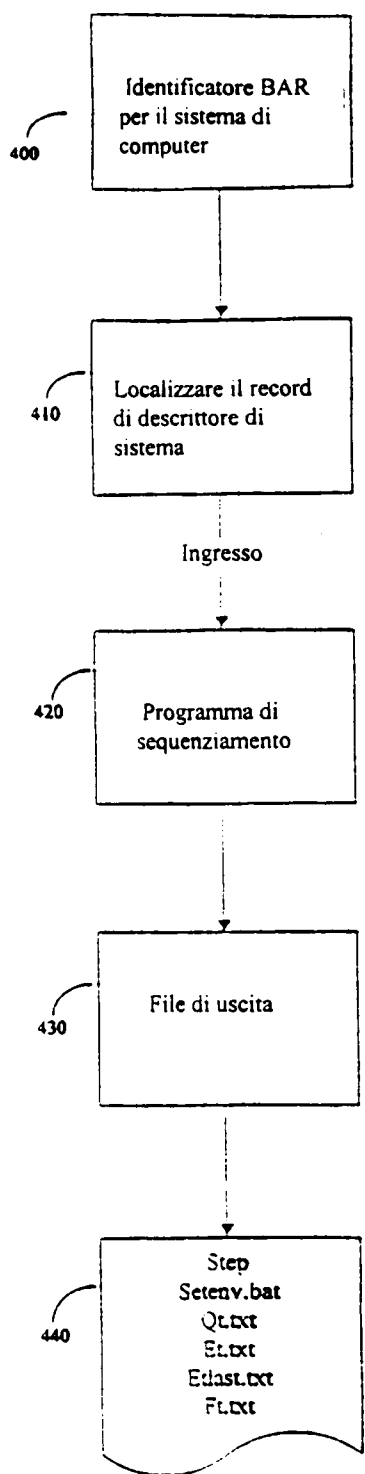


Figura 4

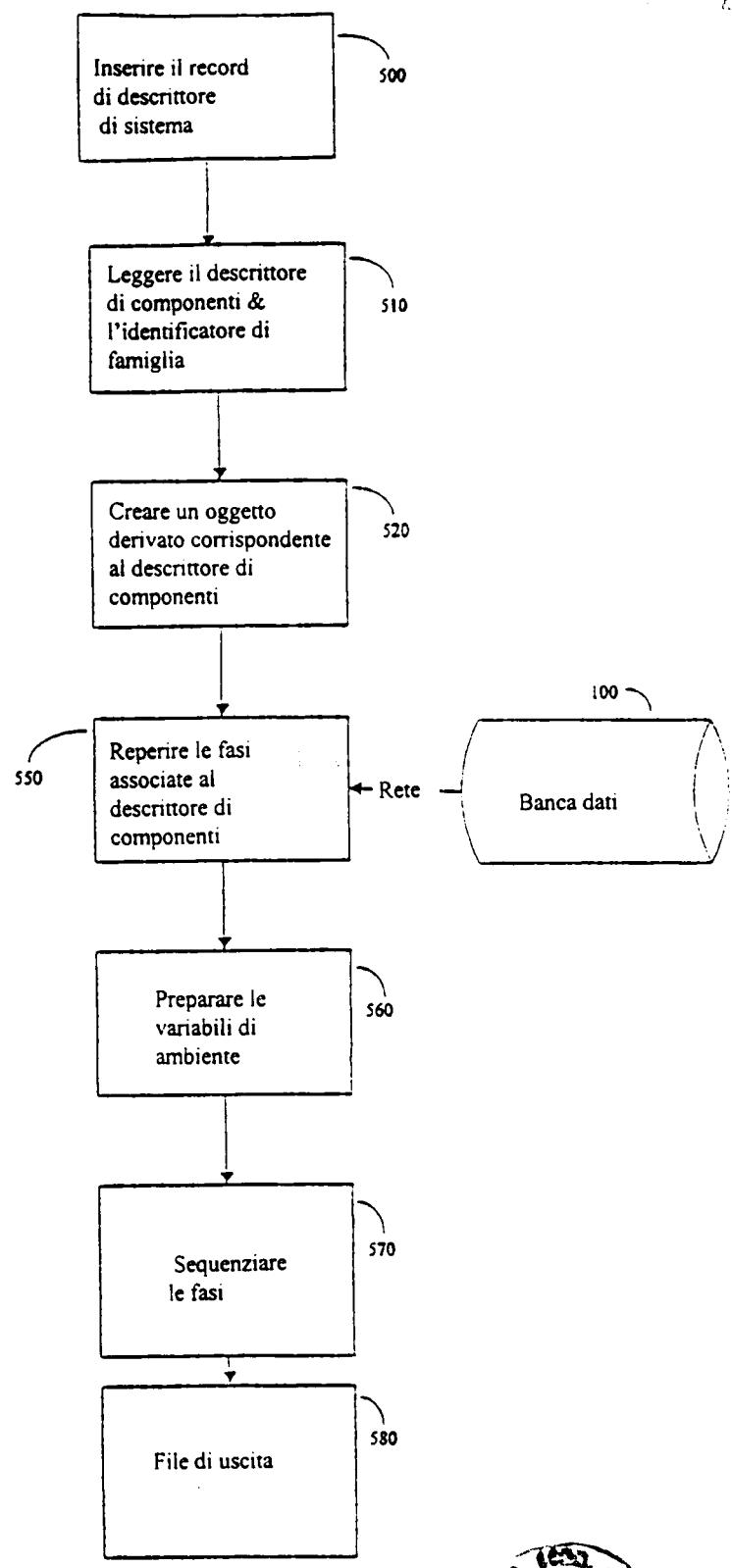


Figura 5



Ing. Franco BUZZI
 Iscritt. ALBO 259
 In proprio e per gli altri

Progetto di banca dati

0802073-0802073

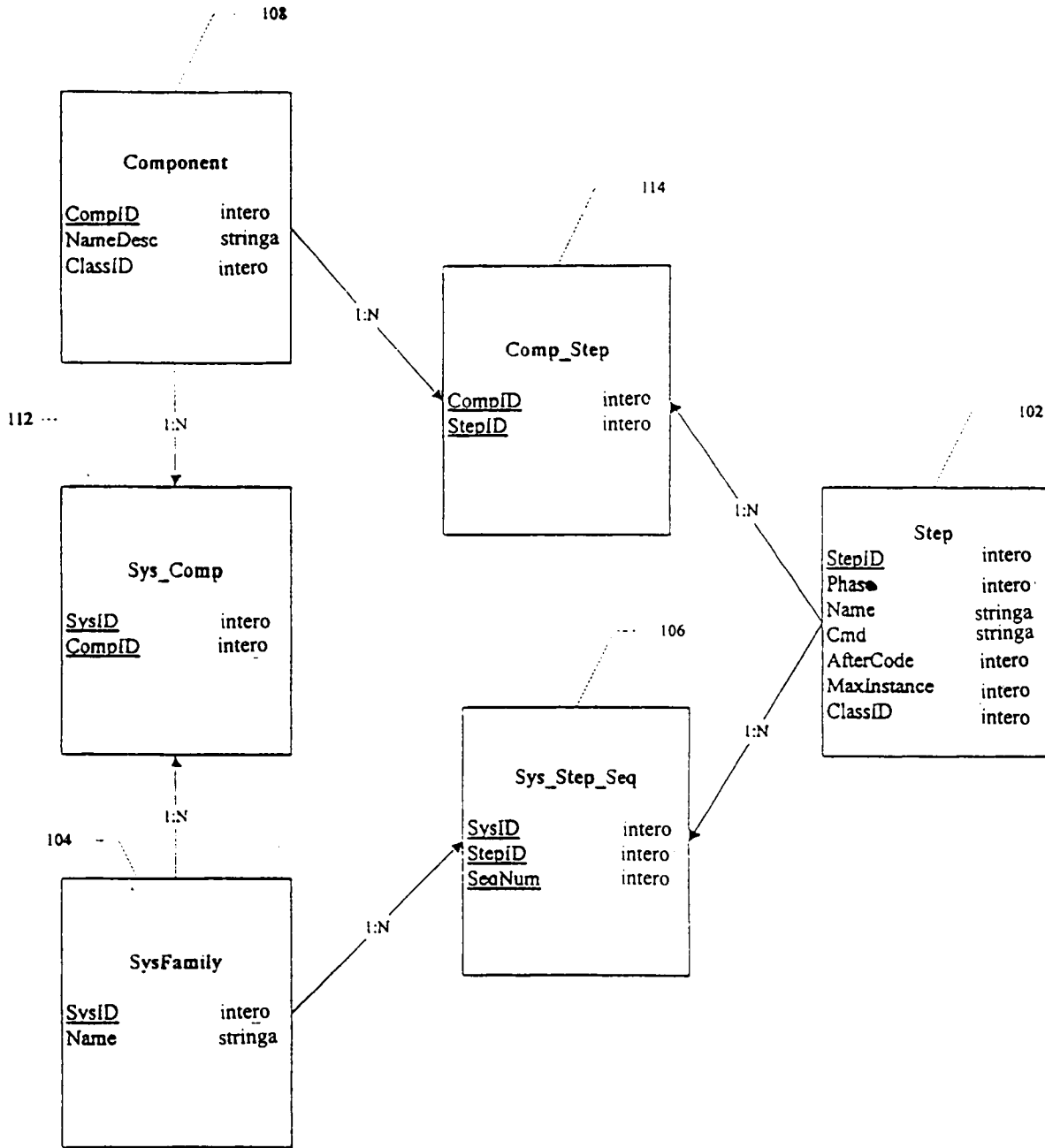


Figura 6



Ing. Franco BUZZI
 INSCRIZ. A.I.S.C. 252
 In proprio e per gli altri

TO 98A 000020

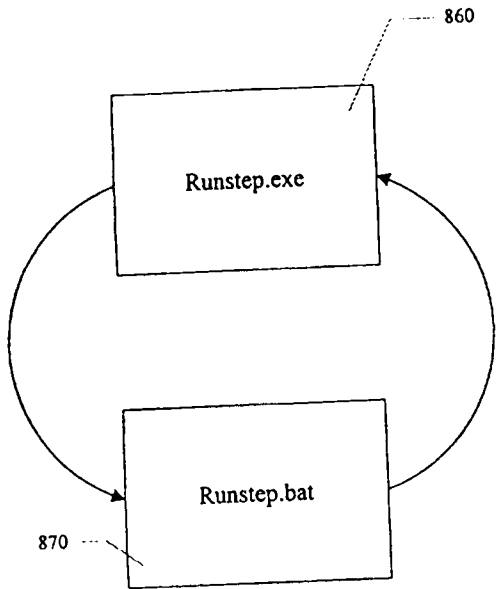


Figure 10

0892073-08997



Ing. Franco BUZZI
Iscriz. ABO 259
Tutti proprio e per gli altri

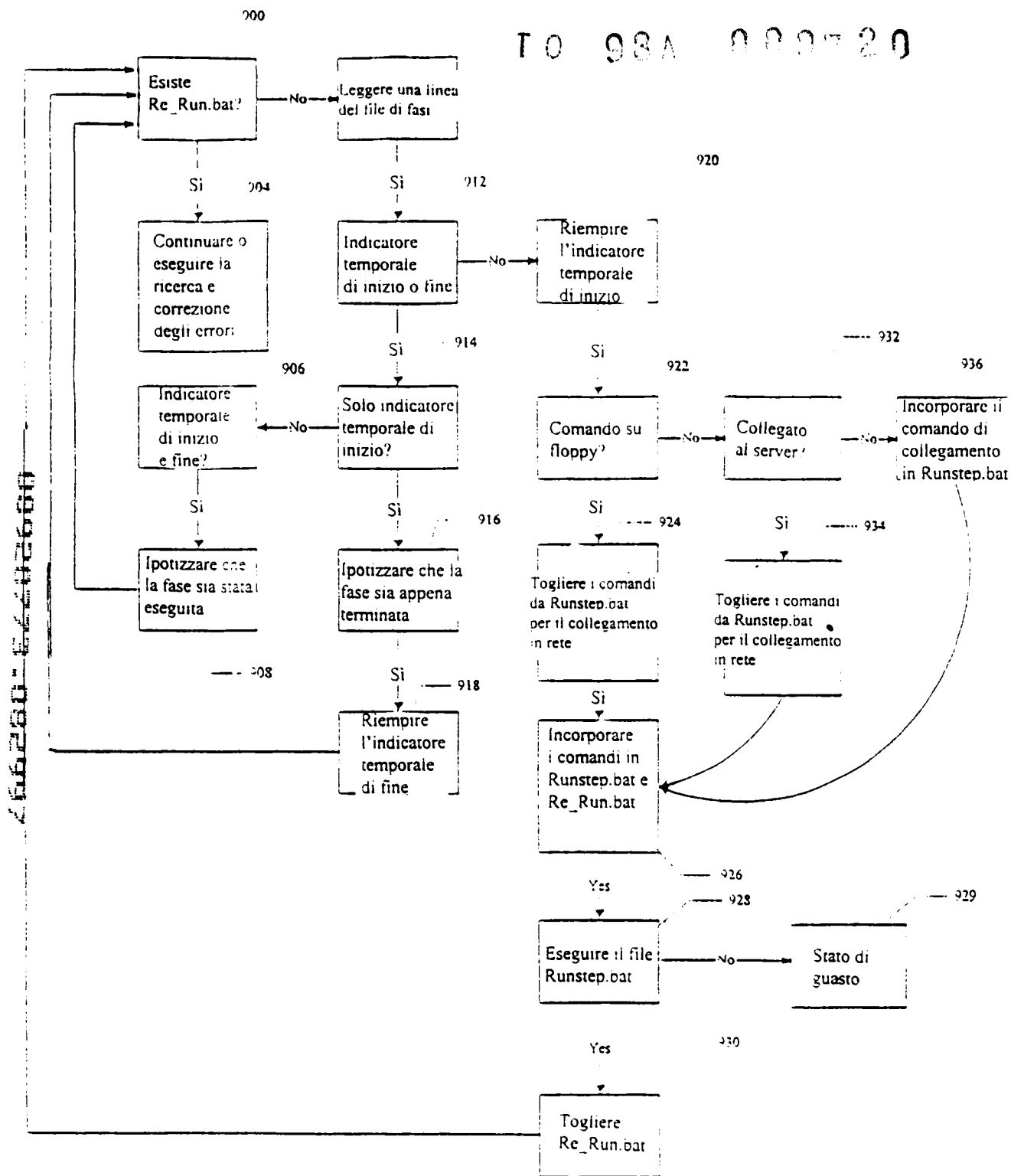


Figura 11