

---

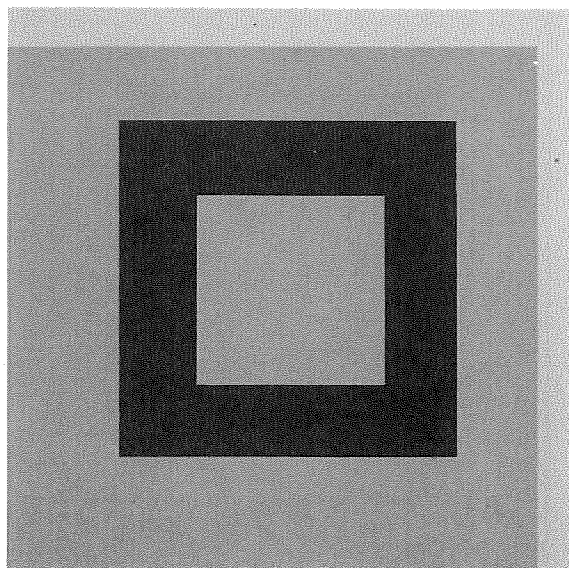
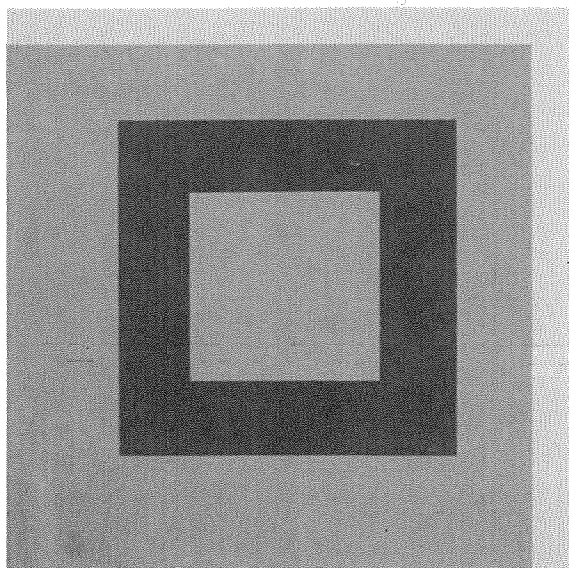
**73**

---

Archivio

# **Cnet: stato della ricerca e proposta 1982**

**N. Lijtmaer**



## Collana Cnet

1. F. Bruni, Gestione degli errori e procedure di correzione
2. R. Marcogliese, Valutazione delle prestazioni di rete locale: metodologie, strumenti, proposte per Cnet
3. C. Montangero, La lista dei requisiti e la struttura dei tipi del linguaggio Ada
4. M. Simi, La rete locale del "Laboratory for Computer Science"
5. V. Ambriola, M. Bellia, P. Degano, Sistemi integrati per la produzione di software
6. G. Attardi, A. Martelli, U. Montanari, Il meccanismo dei moduli nel linguaggio di Cnet
7. U. Montanari, C. Simonelli, On distinguishing Concurrency from Nondeterminism
8. G. Levi, A. Martelli, C. Montangero, Il linguaggio dell'ambiente di programmazione di Cnet
9. P. Degano, R. De Nicola, Una base di dati di moduli in un ambiente integrato per la produzione di software in una rete locale
10. E. Dameri, G. Levi, AISS: un Ambiente Integrato per lo Sviluppo di Software
11. G. Attardi, L'evoluzione degli elaboratori personali: linee di tendenza
12. F. Borgonovo, L. Fratta, A Communication Protocol for Integration of Data and Speech in a Local Communication Network
13. F. Borgonovo, L. Fratta, Design and Performance Evaluation of a Local Communication Subnetwork
14. M. Ajmone, F. Borgonovo, L. Fratta, Considerazioni sulla scelta delle caratteristiche di base della sottorete di comunicazione per Cnet
15. M. Martelli, F. Tarini, Meccanismi di comunicazione "internode": analisi e proposte per una lista di requisiti
16. N. Lijtmaer, Cnet: proposta di ricerca 1981
17. M. Di Santo, L. Nigro, W. Russo, Meccanismi di astrazione del controllo: una proposta per il linguaggio di Cnet
18. M. Ajmone Marsan, Considerazioni preliminari sul protocollo di accesso a canale comune
19. F. Tisato, R. Zicari, Operating System and System Language Requirement List.
20. F. Liguori, G. Rossi, Criteri e meccanismi e modularizzazione
21. E. Dameri, C. Simonelli, Linguaggi intermedi per un ambiente integrato di sviluppo
22. E. Astesiano, E. Zucca, Semantics of CSP via Translation into CCS
23. A. Celentano, P. Della Vigna, Modularizzazione in linguaggi PASCAL-like
24. M. Buttò, Alcune considerazioni sulle reti locali
25. G. Barberis, A. Luvison, Fibre ottiche per comunicazioni ad alta velocità in sistemi ad anello
26. D. Di Pino, Considerazioni preliminari sulla natura e le funzioni di Gateway per una rete locale quale la Cnet
27. S. De Micheli, Integrated Voice, Data, Text and Fax Communications on Local Computer Networks
28. G. Barberis, A. Luvison, G. Pellegrini, G. Pirani, La sottorete di comunicazione nell'ambito del progetto Campus-net
29. G. Fioretto, L. Gabrielli, C. Giuliani, M. Sposini, Proposta per una sottorete di comunicazione in fibra ottica per sistemi distribuiti
30. S. De Micheli, Relazione sulle attività dell'area "Sottosistema di comunicazione"
31. R. Barbuti, M. Bellia, E. Dameri, A. Martelli, C. Simonelli, Gli editor guidati dalla sintassi: strumento base di un ambiente integrato di sviluppo software
32. C. Montangero, A finite state model of the behaviour of an Ada Task

**PROGETTO FINALIZZATO INFORMATICA  
C.N.R.  
Progetto P<sub>1</sub> Cnet**

**N. LIJMAER**

**CNET: STATO DELLA RICERCA E PROPOSTA 1982**

Istituto di Elaborazione dell'Informazione  
CNR - PISA

**ETS/PISA**

## PREFAZIONE

In data 9 Settembre 1982, Il Consiglio di Presidenza del Consiglio Nazionale delle Ricerche, ha approvato la proposta qui riportata quale piano esecutivo 1982 contrattuale, per l'Obiettivo Cnet, Sottoprogetto P1, Progetto Finalizzato Informatica.

La proposta è una sintesi di idee e discussioni avvenute all'interno di Cnet e riflette le linee adottate per il 1982 contrattuale e il 1983 solare.

Tiene inoltre conto dei commenti emersi nelle riunioni degli organi e Comitati del Consiglio Nazionale delle Ricerche.

## 2.P1.CNET. L'Obiettivo Cnet

### 2.P1.Cnet.1. Caratterizzazione dell'Obiettivo Cnet

Cnet si propone di definire le specifiche funzionali e di progetto caratterizzanti l'architettura, il linguaggio di sistema e l'ambiente di programmazione di un sistema funzionalmente distribuito a rete locale e di sperimentare su due prototipi ("test bed") le scelte intraprese.

Le sedi previste per i due prototipi sono l'Istituto di Scienze dell'Informazione dell'Università di Pisa e l'Istituto di Elettronica del Politecnico di Milano. I due prototipi saranno collegati tra loro.

Tra le possibili applicazioni di Cnet sono stati identificati sistemi per automazione degli uffici, sistemi bancari, sistemi per il controllo e la supervisione di centrali telefoniche e sistemi di controllo e supervisione del traffico aereo. Per il loro impatto sociale e di mercato, e per le loro caratteristiche tecnologiche i sistemi per l'automazione degli uffici costituiscono il banco di prova più idoneo su cui concentrare l'interesse per quanto riguarda i prototipi proposti.

Cnet è definito come una federazione di nodi autonomi lascamente connessi e con "hardware" non omogeneo. Cnet è destinato a coprire applicazioni naturalmente distribuite. In ogni applicazione si possono individuare funzionalità che godono di una certa autonomia. Queste funzionalità vengono sviluppate dai singoli nodi autonomi. La disomogeneità dello hardware è connessa alle diverse funzionalità che il sistema deve svolgere. Pur garantendo l'autonomia dei singoli nodi, Cnet si configura come un sistema unico coerente rispetto ad una applicazione.

Questa concezione differenzia l'approccio seguito in Cnet da quello che offre alla applicazione un sistema monolitico centralizzato, pur essendo costituito da componenti fisicamente distribuiti. Questo secondo tipo di sistema è meglio caratterizzato come un sistema multiprocessore senza memoria comune (loosely coupled multiprocessor system) che offre la possibilità di politiche globali per la ottimizzazione dell'uso delle risorse e la trasparenza nell'uso delle stesse, caratteristiche queste di un sistema time sharing. Infatti, Cnet non prevede nessuna politica implicita di utilizzazione delle risorse nè ne consente l'ottimizzazione globale. E' all'interno del singolo nodo autonomo che queste decisioni vengono prese e comunque definite dall'applicazione. I vincoli imposti da applicazioni in tempo reale possono essere soddisfatti solo all'interno del singolo nodo autonomo.

D'altra parte l'approccio seguito in Cnet è diverso da quello tipico di rete di servizio (Arpanet, Ethernet) in cui i singoli nodi sono sistemi indipendenti che comunicano mediante protocolli per scambio di dati. Questi protocolli costituiscono l'unica forma di coerenza presente nella rete. Al contrario, Cnet propone una visione coerente, tramite un unico linguaggio di sistema che incorpora primitive tipiche dei sistemi operativi.

Cnet dovrà garantire la decentralizzazione degli accessi da parte di più utenti, della elaborazione e dei dati. La decentralizzazione geografica è limi-

tata ad un unico ambiente urbano con distanze non superiori a 10Km. Questa limitazione, pur consentendo di coprire applicazioni significative, permette, con tecnologie avanzate ma disponibili, di usare una sottorete di comunicazione fra nodi autonomi con alta banda a basso costo. Questo requisito è essenziale per coprire applicazioni che necessitino di trasmettere oltre che dati, anche voce e facsimile. Nell'ambito dei servizi da fornire per l'automazione dell'ufficio, acquista singolare importanza la possibilità di operare con dati di fonti e caratteristiche diverse integrando i flussi di informazione disponibili. Ne deriva la necessità di garantire un trattamento uniforme ed integrazione funzionale di dati, voce e facsimile.

I moduli di interconnessione (NIM) tra la rete ed i nodi (host) devono essere realizzati con tecnologie a micro-processori e tali da implementare protocolli di comunicazione semplici, eliminando problemi relativi alla compressione della voce. I protocolli dovranno tenere conto della necessità di rispettare vincoli di tipo real-time per il traffico di tipo vocale. La scelta della topologia dovrà consentire sia trasmissione di tipo broadcasting che punto a punto. Dovrà inoltre essere minimizzata la gestione dell'instradamento dei messaggi nella rete.

Il NIM dovrà inoltre prevedere funzioni di "spy monitor" per poter misurare il sistema reale e garantire la progettazione integrata nel software del sistema rete locale del servizio offerto dal "monitoring centers" per la misura, il debugging, la comunicazione, controllo di guasto ecc.

In Cnet, i nodi potranno essere disomogenei sia come hardware che come configurazione, pur rientrando in una fascia che va dal mini-calcolatore al "personal computer". Questi calcolatori dovranno essere disponibili sul mercato e la loro scelta sarà funzione del tipo di applicazione a cui vengono dedicati. Cnet garantisce la comunicazione con altre reti locali o con la rete pubblica tramite un gateway.

Il linguaggio di Cnet deve consentire la programmazione di sistemi distribuiti e deve possedere costrutti tali da aumentare l'affidabilità del sistema quali tipi, astrazioni, moduli, meccanismi di protezione e di fault-tolerance. Deve inoltre garantire la massima verificabilità statica. Deve consentire la definizione di qualunque politica di gestione delle risorse. Nessuna strategia di schedulazione è imposta dal linguaggio. E' possibile programmare, all'interno di un singolo nodo autonomo, applicazioni in tempo reale stretto.

La visibilità del sistema offerta dal linguaggio è completa. Non esiste nessuna possibilità di virtualizzazione globale delle risorse della rete. Il programmatore-sistemista è in grado di controllare con il linguaggio, l'allocazione dei programmi e le risorse coinvolte, e dunque di definire per la sua applicazione una particolare rete virtuale ed i nodi virtuali che la compongono. Il linguaggio permette inoltre di specificare l'allocazione dei nodi virtuali sui nodi fisici. Ne deriva che il linguaggio contiene come sottoinsieme il linguaggio di comandi e configurazione. Sono previste, dunque, primitive per collegamento, caricamento, gestione delle risorse, lancio di un programma ecc.

Il linguaggio deve essere facilmente portabile, sui diversi nodi, anche eterogenei, che compongono la rete. La portabilità è garantita dall'esistenza di un linguaggio intermedio che permette di presentare una interfaccia uniforme verso il sistema di programmazione nonostante l'eterogeneità dei nodi (target). Garantisce la possibilità di rendere ortogonali le evoluzioni dell'hardware e

del software. La macchina astratta definita dal linguaggio intermedio prevede oltre all'interprete sequenziale un nucleo per l'implementazione della concorrenza all'interno del nodo virtuale sia esso uni-processor o multi-processor con memoria comune; un nucleo per l'implementazione dei meccanismi di comunicazione inter-node; un nucleo per la configurazione dell'intero sistema distribuito ed infine un nucleo per la gestione dei malfunzionamenti che rendono non disponibili le risorse fisiche del sistema ("crash").

L'ambiente per lo sviluppo di software (AISS) è un ambiente multi-user distribuito. La distribuzione non consiste semplicemente nel collegamento di varie stazioni di lavoro, ma fa riferimento anche a funzionalità diverse tipiche dell'organizzazione del lavoro del software (per es. le attività di management, definizione di sistemi, programmazione). A funzionalità diverse corrispondono, in generale, nodi diversi; tali funzionalità, inoltre, hanno un impatto diretto sulla definizione di strumenti distribuiti che garantiscono la cooperazione. Tali strumenti comprendono la gestione di servizi comuni, la comunicazione, la gestione di basi di dati: AISS è un'applicazione distribuita caratterizzata da una accentuata autonomia dei nodi, e, in quanto tale, può essere considerata una naturale applicazione di Cnet. Nell'utilizzazione come "software factory" o come "campus net", mette chiaramente in evidenza i vantaggi di una soluzione distribuita (affidabilità, espandibilità, assenza di decadimento delle prestazioni in fasi di saturazione) rispetto ad una tradizionale soluzione concentrata.

AISS è un insieme integrato ed estendibile di strumenti granulari (cioè, con funzionalità semplice e tra loro componibili). AISS, inoltre, è ricordato all'unico linguaggio di programmazione ad alto livello, che è anche il linguaggio di comandi per l'interazione con il sistema. Gli strumenti stessi sono programmati in tale linguaggio. Questa connessione, già presente in alcuni sistemi di programmazione LISP e nei requisiti dell'ambiente di ADA, risulta ulteriormente rafforzata dal fatto che il linguaggio permette la gestione esplicita delle risorse e non si basa, quindi, su un sottostante sistema operativo. E' questa una caratteristica importante per AISS, la cui portabilità è solo legata alla portabilità del linguaggio. Questa stessa caratteristica, unita a quella della granularità, permette di vedere il supporto run-time di una applicazione su target come un opportuno sottoinsieme dello stesso AISS. Ciò permette lo sviluppo completo della applicazione su host, senza richiederne una simulazione del target. D'altra parte, in funzione della complessità e delle necessità di manutenzione-modifica-riconfigurazione di una applicazione embedded, lo stesso target può sostenere un opportuno sottoinsieme di strumenti.

AISS prevede strumenti di supporto alla programmazione distribuita. Particolarmente importanti sono, a questo riguardo, strumenti che permettano la configurazione e riconfigurazione di applicazioni, strumenti che permettano il debugging, e infine strumenti di valutazione, misurazione e testing a run-time.

AISS fornisce un'interfaccia altamente interattiva, in cui risultano essenziali dispositivi di interazione grafica anche sofisticati. L'interattività da un lato permette di sfruttare appieno le caratteristiche di integrazione offerte da AISS, dall'altro, richiedendo dispositivi di interazione complessi, fornisce una ulteriore motivazione per la soluzione distribuita, in quanto si

renderebbe comunque necessario portare più potenza di calcolo sui terminali.

AISS fornisce un supporto anche alle attività di tipo organizzativo ed amministrativo attraverso strumenti per la gestione di archivi e strumenti di comunicazione. Da questo punto di vista, AISS dovrà comprendere un sistema di automazione di ufficio, con strumenti che prevedano, ad esempio, una posta elettronica, la preparazione di documenti, la gestione dell'agenda, della corrispondenza, delle telefonate etc. A questi si aggiungeranno strumenti di gestione più specifici, fondamentalmente strumenti di comunicazione selettivi e specializzati e di interazione fra diverse basi di dati.

Nell'ambito del progetto, si intende assegnare un ruolo importante alla definizione formale del linguaggio (sintassi astratta, semantica statica, semantica dinamica), sia per ragioni metodologiche che come base per l'implementazione degli strumenti. Gli aspetti su cui si focalizzerà l'attenzione sono naturalmente quelli connessi alla semantica del parallelismo e del non-determinismo.

Si prevede una verifica delle scelte architetture, complessive di Cnet mediante lo sviluppo d'un sistema integrato di automazione degli uffici, dato che, questi sistemi costituiscono un'applicazione naturale per le architetture lascamente connesse e funzionalmente distribuite. Infatti i granuli della Unità di Elaborazione corrispondono a funzioni autonome che possono essere eseguite in parallelo con altre funzioni. Vedasi per esempio le funzioni relative alla preparazione dei documenti, alla gestione dei documenti, ecc. Le Unità di Elaborazione sono eterogenee e si adattano ai vari servizi a seconda della possibile elaborazione all'interno dell'ufficio. I riferimenti ai dati godono della proprietà di località, e cioè deve esistere la possibilità di distinguere il trattamento dei dati locali da quello di dati remoti. La frequenza di sincronizzazione tra le diverse funzioni e la quantità di informazione trasmessa costituiscono infine parametri significativi sia nel progetto della rete locale, sia della applicazione. Questi parametri sono significativi per esempio nel progetto di Posta Elettronica, Agenda, "Files Servers" ecc.. Inoltre la potenzialità di integrare dati provenienti da diverse fonti come: messaggi, voce e facsimile costituisce una possibilità offerta dalle scelte architetture di Cnet (la sottorete di comunicazione e i relativi protocolli prevedono un trasferimento uniforme) ed è la premessa di un sistema integrato di automazione degli uffici. La scelta come Host di mini-calcolatori o "Personal Computers" rafforza la possibilità di esplorare un'attività, più o meno complessa, ma autonoma, ad ogni unità di elaborazione. Infine alcuni granuli di strumenti previsti nel AISS (ad esempio Editors) costituiscono a loro volta granuli dei sistemi interattivi che permettono una interfaccia uniforme tra l'utente e il sistema di automazione degli uffici.

Si noti tuttavia che l'attività nel campo dell'automazione degli uffici che si intende portare avanti nell'ambito del Cnet riguarda principalmente gli aspetti sistemistici e di comunicazione e non fa riferimento ad aspetti architetture delle singole "Work Station", nè alle apparecchiature periferiche specializzate.



## 2.P1.Cnet.2. Stato della ricerca

L'attività del periodo I° Settembre 1980 - I° Settembre 1981, si è impostata su quattro linee di ricerca, che oltre a garantire la necessaria articolazione del progetto hanno verificato e precisato gli obiettivi che Cnet si era preposto ossia le specifiche funzionali caratterizzanti l'architettura, il linguaggio di sistema e l'ambiente di programmazione di un sistema funzionalmente distribuito a rete locale.

2.1. Nell'ambito della prima linea relativa alla sottorete di comunicazione Cnet sono da segnalarsi i seguenti sviluppi.

2.1.1. Definizione di una struttura topologica e di un protocollo di accesso per la Sottorete di Comunicazione, in grado di rispondere adeguatamente ai requisiti che sono stati individuati nell'attività passata dal gruppo di lavoro per la sottorete Cnet. I principali requisiti si possono riassumere brevemente come segue:

- Topologia in grado di connettere stazioni la cui distanza può raggiungere l'ordine di qualche Km. (2 a 10)
- Accesso garantito a un numero di stazioni dell'ordine di qualche centinaio. (256)
- Possibilità di integrare insieme ai dati, anche il traffico vocale e tutti gli altri tipi di traffico previsti nelle applicazioni note come Office Automation. (Videotext)
- Capacità di collegamenti Broadcast.
- Meccanismo di accesso e controllo della sottorete completamente distribuito.

La struttura proposta, nota come EXPRESS-NET, (in fase di brevetto tra CNR e Stanford University) consiste di un bus unidirezionale, realizzabile su doppio, cavo coassiale, guida d'onda o fibre ottiche. In tale bus sono individuabili due parti:

- L'Outbound channel o canale di trasmissione cui tutte le stazioni sono collegate per la trasmissione.
- Inbound channel o canale di ricezione cui tutte le stazioni sono collegate per la ricezione.

La peculiarità del protocollo di accesso base EXPRESS consiste nella capacità di creare dei "treni" di pacchetti trasmessi dalle varie stazioni in modo tale che:

- A ciascuna stazione è garantita la possibilità di trasmettere un pacchetto di informazioni su ciascun treno in modo ordinato. Questa proprietà è fondamentale dal punto di vista della correttezza nell'accesso. Infatti benché in ciascun treno i pacchetti siano ordinati allo stesso modo in cui le stazioni sono ordinate lungo l'Outbound channel, limitando la lunghezza massima del pacchetto si ottiene un meccanismo di priorità "Round/Robin", e un ritardo di

accesso massimo garantito, cosa basilare se si vuole trasmettere traffico in tempo reale come la voce.

- I pacchetti vengono trasmessi in modo tale che la distanza tra due pacchetti consecutivi in un treno è dell'ordine di qualche bit. Questo fatto consente un'altissima efficienza nell'utilizzo della banda disponibile indipendentemente dalle dimensioni della rete, dal numero di stazioni, dalla lunghezza del pacchetto e dalla banda stessa.
- L'istante di trasmissione del pacchetto in un treno viene determinato in modo totalmente distribuito. Ciascuna stazione determina tale istante semplicemente in base all'informazione che trae istantaneamente dall'Outbound channel. Non è necessaria quindi memoria del passato. In particolare una stazione può trasmettere solo quando vede transitare nell'Outbound channel la fine di un altro pacchetto trasmesso a monte. Nel contempo ascolta ciò che viene da monte e se avverte l'inizio di un'altra trasmissione, interrompe la propria.
- Treni successivi sono iniziati da ciascuna stazione sulla base dell'osservazione istantanea dell'Inbound channel. La distanza tra due treni consecutivi viene garantita entro un massimo limitato dell'ordine del tempo di propagazione "end to end". Allo scopo di semplificare al massimo l'operazione di partenza di nuovi treni si è proposto un collegamento a "Z" fra Inbound e Outbound channels.

Quanto sopra esposto permette di ottenere un accesso ordinato e sincrono, necessario per la voce nella configurazione di carico massimo, il tutto senza necessità di affidarsi a orologi locali, centrali, o a scambio di informazioni di sincronismo tra le stazioni cosicchè la banda è praticamente disponibile per la trasmissione dell'informazione.

L'implementazione software di tale protocollo appare estremamente semplice e infine occorre notare che non viene richiesto nessun particolare algoritmo di recovery del protocollo in caso di cadute di una o più stazioni o parte della sottorete.

Una variante al protocollo EXPRESS, è stata proposta. La comunicazione avviene sulla base di "treni" la cui formazione è basata sugli stessi meccanismi di EXPRESS, però il treno dati viene iniziato in modo asincrono. Ogni stazione ha, a differenza di EXPRESS, tre connessioni sul "bus". Infatti questo protocollo prevede la conoscenza dei tempi di propagazione fra trasmettitore e ricevitore.

Uno studio qualitativo sulla "fault tolerance" di alcuni protocolli mette in evidenza la consistenza dei protocolli CSMA - CD ed EXPRESS-net senza però quantizzarne l'impatto sulla robustezza. Una valutazione del protocollo CSMA - CD per trasmissione voce ripropone però uno scenario diverso da Cnet.

2.1.2. Analisi della complessità realizzativa di certe strutture della rete con diversi mezzi trasmissivi. In particolare, si sono presi in considerazione possibili realizzazioni in Fibra ottica e in Cavo Coassiale, sia della topologia e i protocolli proposti da EXPRESS che di una topologia a due livelli con protocolli CSMA - CD modificati e ALOHA. Questa ultima proposta fatta allo scopo di migliorare le prestazioni di costo - affidabilità di una struttura in fibra prevede due livelli: una struttura principale a bus bifibra monodirezionale che connette nodi attivi allocati negli edifici; ed una struttura secondaria a stella con il centro stella in corrispondenza dei nodi attivi collegati dal bus, destinata a servire le stazioni all'interno degli edifici ad una distanza di qualche centinaio di metri.

Il centro stella è costituito da un accoppiatore ottico passivo più un rigeneratore in grado di concentrare sedici ingressi.

Con questa struttura il costo per stazione si aggira intorno a 1ML. Per quanto riguarda l'affidabilità si vede che è circa dimezzato il numero di rigeneratori attivi in serie al bus rispetto alla struttura EXPRESS-NET in fibra.

Per quanto riguarda i protocolli di accesso che una tale struttura è in grado di supportare, abbiamo:

- il CSMA - CD che però non è realizzabile, dati i requisiti richiesti dallo scenario Cnet che prevedono un tempo di propagazione maggiore del tempo di trasmissione di un pacchetto.

Nell'ambito dei protocolli ad accesso casuale l'unico direttamente utilizzabile è l'ALOHA che però, data la scarsa efficienza, richiederebbe una velocità in linea di almeno 100 Mb/s.

- l'algoritmo di accesso EXPRESS può essere modificato per poter funzionare su questa struttura, dovuto al fatto che nella stella non esiste un naturale ordinamento delle stazioni. Con tale modifica il protocollo presenta un'efficienza che dipende dalla lunghezza dei rami della stella. Per esempio con un ramo stella di 200 m l'efficienza si riduce al 33% con pacchetti di 500 bit, mentre con un ramo di 20 m l'efficienza risulta pari all'80%.

In una tale struttura pertanto la stella deve essere utilizzata solo per servire aree limitate tenendo quindi ad una struttura lineare con concentramento degli spillamenti per motivi di costo.

Per quanto riguarda la realizzabilità di un prototipo a fibre è stato studiato e sperimentato in laboratorio un connettore che permette l'innesto di 19 fibre in una.

Studi sul costo e affidabilità di EXPRESS in relazione ad una possibile realizzazione in Fibra ottica e in Cavo Coassiale hanno portato alle seguenti considerazioni.

- FIBRA - Bus con accoppiatori ottici passivi unidirezionali.

Tali accoppiatori presentano attualmente una notevole perdita di potenza e pertanto è necessario inserire in serie alla fibra frequenti stadi di amplificazione e rigenerazione del segnale ottico.

Nella migliore delle ipotesi è necessario uno stadio attivo ogni 6-8 inserzioni passive.

Il costo di una tale struttura per stazione risulta alquanto elevato e dell'ordine di 6ML.

Esistono inoltre notevoli problemi di affidabilità dovuti a due ordini di motivi: la bassa affidabilità e stabilità dei componenti ottici disponibili; e l'elevato numero di tali componenti posti in serie alla linea. I ripetitori attivi raggiungono infatti l'ordine di almeno un centinaio. Ciò complica anche i problemi di alimentazione, manutenzione, ecc.

- Cavo - Bus con accoppiatori passivi.

Le due soluzioni analizzate prevedono accoppiatori passivi unidirezionali mediante trasformatori o accoppiatori passivi bidirezionali (tipo ETHERNET) con sensing unidirezionale. A tale proposito si noti che il protocollo EXPRESS non necessita della unidirezionalità della trasmissione bensì della capacità di riconoscere, da parte di ogni stazione, sull'out-bound channel se il segnale in transito proviene da monte o da valle.

Nel primo caso la perdita di inserzione è tale che si può pensare di mettere un rigeneratore attivo in serie al cavo ogni 20-40 spillamenti, oppure ogni 200 m.

Con una struttura che prevede accoppiatori passivi bidirezionali con sensing unidirezionale si può pensare di inserire un ripetitore attivo ogni 50 spillamenti circa (od ogni 200-300 m).

Tale soluzione rispetto alla prima ha il vantaggio di poter utilizzare gran parte dei componenti già sviluppati per reti tipo ETHERNET.

Le valutazioni di costo basate sulle esperienze di realizzazione di reti simili a ETHERNET sono dell'ordine dei 300 KL per ogni stazione.

Ulteriori studi dovranno essere svolti onde poter fornire le specifiche funzionali della sottorete di comunicazione di Cnet in fibra.

2.1.3. I moduli di interconnessione (NIM) sono preposti ad interfaccia terminale di vario tipo (test computers, terminali telefonici, terminali facsimili etc.) al mezzo trasmissivo, da realizzare in fibre ottiche o in cavo coassiale a velocità superiore a 34 MBps. Il progetto del NIM non comprende la realizzazione del tranceiver, unità per la transcodifica dei segnali da e verso la rete, e quindi è sganciata dalla tecnologia per esso adottata, dipendono però dalla velocità scelta, per la ricetrasmissione.

Le funzioni già individuate nel corso degli studi prevedono che il NIM svolga funzioni di ricetrasmissione delle informazioni, in accordo con il protocollo EXPRESS, della loro bufferizzazione e di moltiplicazione dei nodi virtuali, curando il protocollo di scambio con gli host computers. Esso realizzerà quindi in parte il livello fisico dei protocolli, il livello data link ed il livello network. Non è escluso, l'inclusione nel NIM del livello transport per l'interfacciamento di alcuni terminali. Sono infine previste alcune attività di monitoraggio del sistema atte a supportare funzioni di controllo del flusso, diagnostica etc.

In particolare la logica dello "spy monitor", cioè tutta la parte che svolge funzione di confronto e riconoscimento sui campi del pacchetto deve essere sufficientemente veloce da poter analizzare tutti i pacchetti transitanti sulla rete. Il tipo di confronti che si possono proporre sono: misura della lunghezza del campo dati dei pacchetti per la generazione di report sulla distribuzione delle lunghezze dei pacchetti, confronto del campo attività da o per una data stazione o gruppo di stazioni (multicast), misure di interesse specifico del protocollo EXPRESS. L'efficienza di tale componente è vitale per lo strumento finale.

2.2. La seconda linea di ricerca riguarda il linguaggio di sistema di Cnet che propone una visione di un sistema unico ma distribuito e che incorpora le primitive tipiche dei sistemi operativi.

Si è escluso in partenza la idea di definire un nuovo linguaggio perchè non permetterebbe nè il confronto con altri programmi di ricerca nè una diffusione dei risultati, in quanto singolari. Dopo aver affrontato l'analisi dei requisiti per il linguaggio di Cnet, e avendo preso in rassegna quelli che sono tra i linguaggi noti i più idonei a risolvere il problema della programmazione

distribuita così come emergeva dai requisiti elaborati (PASCAL, MODULA, CONCURRENT PASCAL, CSP, PLITS, CLU e ADA), la scelta è caduta su ADA. Oltre alle ragioni tecniche di seguito elencate, e anche se ADA non assolve a tutte le proprietà desiderabili, la scelta è confortata dalla previsione di ampia diffusione di ADA e della possibilità di interazione con una vasta comunità internazionale che attualmente lavora intorno ad ADA sia scientifica che industriale. Infatti oltre ai numerosi programmi di sviluppo di compilatori, particolare interesse è riposto sull'ambiente di programmazione ricordato ad ADA.

Il linguaggio ADA, oltre alle strutture di controllo dei linguaggi algoritmici ed alla possibilità di definire tipi, con capacità di controllo forte sui tipi, possiede il costrutto Package che realizza il concetto di modulo. Detto costrutto è un meccanismo base per incapsulare oggetti da proteggere, per supportare astrazione sui dati, ed è l'unità di compilazione con capacità di verifica statica della consistenza delle interfacce.

Il costrutto task permette di definire processi concorrenti. Ciascun task è dotato di un proprio ambiente locale e può comunicare e sincronizzarsi con altri task tramite i comandi di entry-call e di accept, cioè con un meccanismo simile alla chiamata a procedura remota. Inoltre ADA fornisce una struttura di controllo, detta "selective-wait", che permette di esprimere il concetto di non-determinismo e di ricezione selettiva per condizione.

La programmazione di sistema è avvantaggiata dalla possibilità di accedere a parametri dipendenti dal sistema e di avere un preciso controllo della rappresentazione fisica dei dati.

Per venire incontro ad esigenze di real-time, ADA offre anche meccanismi di time-out, di gestione delle eccezioni e dà la possibilità di assegnare priorità (statiche) ai task, in modo da influenzare, in una certa misura, le politiche di scheduling insite nel supporto run-time del linguaggio.

Ciò nonostante, un'analisi approfondita di ADA come linguaggio per la programmazione in tempo reale ha messo in evidenza alcune limitazioni. Infatti dopo aver definito primitive di macchina astratta per la gestione di problemi in tempo reale stretto si intende proporre alcune modifiche ad ADA che lo rendano idoneo a dare completa visione e programmabilità dello Scheduler da parte dell'utente, senza però alterare la semantica della concorrenza.

La portabilità è garantita dal linguaggio intermedio di Cnet (X-code) che garantisce la omogeneizzazione di tutti gli hardware disomogenei, e sarà usato come oggetto per il compilatore del linguaggio.

La sua definizione coincide con la definizione della architettura della macchina astratta Cnet di basso livello.

Attualmente sono stati definiti i seguenti elementi:

- . Una architettura elementare, e filosofia uniprocessor, basata sul concetto di attività sequenziale; a questo livello è definito un meccanismo elementare di commutazione di contesto, sia sincrona (operazione "switch") che asincrona ("eventi asincroni"). Il comportamento della macchina "di livello" è completamente deterministico e non esistono politiche built-in.
- . Una tecnica generale per la costruzione di livelli superiori di macchina astratta, partendo dal livello di X-code.
- . Si è estesa la macchina base con il concetto di canale di comunicazione (sistematizzazione del concetto di I/O) e con quello di ambiente di controllo

(sistematizzazione del concetto di "stato supervisore"). Servendosi di tali concetti si è formulata una definizione esemplificativa di come, usando canali di comunicazione fra ambienti chiusi e ambienti di controllo, possa essere costruito un sistema a nodi virtuali.

Per quanto riguarda la "fault-tolerance" i risultati possono essere schematicamente riassunti nel seguente modo:

- . Analisi di un adeguato meccanismo di gestione delle eccezioni per trattare tutte le condizioni anomale prevedibili dei componenti software (forward error recovery) /Collana Cnet N. 42/.
- . Analisi di opportuni meccanismi di "backward error recovery" per la gestione di crashes, (malfunzionamenti che rendono non disponibili risorse fisiche del sistema).
- . Uso di meccanismi di astrazione del controllo per implementare meccanismi di recovery per errori software non previsti.

2.3. Per quanto riguarda l'Ambiente Integrato per lo Sviluppo di Software, è stata definita l'organizzazione generale di un ambiente, basato su una concezione di integrazione fra ambiente su macchina ospite e ambiente su target, legato al linguaggio ADA, e comprendente i seguenti componenti.

#### 2.3.1. Ambiente su macchina ospite

- . Nucleo (KAPSE), avente lo scopo di garantire la portabilità degli strumenti, realizzando, per esempio, l'indipendenza dal sistema operativo. Contiene tutto ciò che è richiesto per lo sviluppo degli strumenti e non è direttamente fornito dal linguaggio (supporto a tempo di esecuzione). Tipicamente, fornisce le primitive per definire una base di dati di progetto, per comporre ed invocare strumenti (anche dinamicamente), l'interfaccia verso l'utente, etc.
- . Strumenti che operano sulla rappresentazione dei programmi in sintassi astratta (ad alto livello), e forniscono un supporto alla creazione, modifica di programmi ed alla verifica del loro comportamento per tutti gli aspetti che non dipendono dal reale ambiente operativo. Gli strumenti coinvolti comprendono l'editor, il parser, il prettyprinter, vari analizzatori sintattici e semantici, il linker ad alto livello, l'interprete della sintassi astratta, etc. E' stata proposta una metodologia di sviluppo degli strumenti che è basata sulla definizione formale del linguaggio (sia sintattica che semantica e permette di ottenere in modo sistematico strumenti granulari con funzionalità ben definita e di ridurre il costo dello sviluppo di ambienti per linguaggi diversi.
- . Linguaggio di comandi, con relativo interprete, che fornisce l'interfaccia interattiva verso l'utente, e garantisce l'estendibilità dell'ambiente.
- . Compilatore verso la macchina intermedia che caratterizza e unifica i nodi.

### 2.3.2. Ambiente su macchina target

- . Strumenti per il debugging simbolico, la valutazione e la misura nell'ambiente operativo.
- . Linker-loader, direttamente legato ai costrutti di configurazione presenti nel linguaggio di sistema.
- . Strumenti per l'integrazione e la comunicazione tra strumenti su macchina ospite e strumenti su target.

E' opportuno rilevare come le caratteristiche di distributezza influenzino almeno in prospettiva, l'ambiente per due aspetti. Essendo la macchina ospite distribuita, aspetti relativi alla distribuzione sono presenti nel KAPSE (basi di dati, comunicazione tra strumenti). Il fatto che il linguaggio dovrà permettere la definizione di programmi distribuiti, introduce aspetti di distribuzione negli strumenti ad alto livello (linker ed interprete) e in tutti gli strumenti su target.

2.4. La quarta linea di ricerca è nata sulla constatazione che non esistevano validi metodi di specifica semantica per sistemi con concorrenza e ancor meno per sistemi con vincoli di "real-time".

Se dunque si vuole documentare il sistema Cnet con la specifica rigorosa delle sue componenti più innovative, non è possibile applicare metodologie note e collaudate, ma occorre invece cercare e proporre strumenti adeguati, che serviranno poi non solo per la documentazione finale, ma anche per controllo della progettazione in fase di specifica del prototipo.

Mentre era in attuazione la definizione della architettura software di Cnet, ci si è concentrati sulla ricerca di metodologie da applicare alla produzione delle specifiche semantiche e quindi su attività di supporto, tenendo però conto dei requisiti già precisati e quindi in relazione alla scelta di ADA come linguaggio Cnet.

Si è individuata in sistemi tipo CCS (e variazioni) una proposta valida per la specifica di parti di Cnet. Tale approccio è stato in particolare testato su linguaggi importanti e noti come i CSP di Hoare, i DP di Brinch Hansen e il metalinguaggio usato da Bjorner e Folkjaer per descrivere la semantica dell'ADA.

In particolare cercando delle direttive per semantiche di traduzione, presso l'unità di Genova è stata sviluppata una proposta di metodologia di descrizione semantica (MHA, Modular Hierarchical Analysis), attualmente in fase di testaggio su vari sottoinsiemi significativi dell'ADA.

### 2.5. Documentazione

L'Obiettivo Cnet stampa regolarmente una collana di documenti che, oltre ad informare gli interessati dell'evoluzione dell'Obiettivo, è veicolo di discussione attraverso cui ogni componente di Unità Operativa o Gruppo di Lavoro esprime i suoi contributi tecnico-scientifici.

Attualmente, la Collana Cnet è stata stampata fino al numero 40.

## 2.P1.CNET.3. Aggiornamenti al piano esecutivo 1981

### 3.1. Sottorete di Comunicazione Cnet

Per quanto riguarda la scelta fra cavo e fibra, gli studi già fatti presentano dati abbastanza generici presi soprattutto dalla tecnologia telefonica e quindi non sempre significativi per la rete Cnet. Ad esempio in Cnet non interessa molto il passo di ripetizione nè vedere che la fibra può permettere tratte maggiori. Infatti nelle topologie considerate non si hanno mai trasmissioni a lunga distanza ma piuttosto sono significativi i costi delle inserzioni e spillamenti. Sono questi che possono determinare perdite e quindi necessità di amplificazione. Il costo dei ripetitori a 34 Mbit/s di circa 3ML è dieci volte superiore a quelli utilizzati in strutture tipo ETHERNET. Queste differenze rendono le scelte molto incerte.

Si ritiene dunque necessario un ulteriore studio da parte della Unità Operativa CSELT, volto a fornire il supporto necessario alla definizione delle specifiche della parte trasmissiva della sottorete di comunicazione di Cnet in fibra ed alla definizione di un prototipo sperimentale atto a dimostrare la fattibilità delle scelte sistemistiche operate. Il prototipo non dovrà però necessariamente sottostare agli stessi vincoli di economicità, affidabilità e dimensionamento del sistema di cui si vuol dimostrare la fattibilità. Le specifiche del prototipo dovranno inoltre comprendere una valutazione delle risorse necessarie per la sua eventuale realizzazione.

Parallelamente si considera che le tecnologie a cavo, offrono un vantaggio non trascurabile ai fini del progetto, che è quello di permettere tempi brevi di realizzazione di un prototipo. Questo elemento sommato al fatto che il rapporto di costo dei componenti è di 1 a 10 in favore del cavo, che l'affidabilità è estremamente critica nella fibra, e che la sottorete in cavo sembra realizzabile utilizzando componenti già sviluppati ed ingegnerizzati per reti esistenti, ha permesso di stabilire la seguente proposta operativa.

- Sviluppo del progetto per la realizzazione del bus unidirezionale via cavo allo scopo di verificare completamente la fattibilità delle soluzioni ipotizzate.

Tale studio non dovrebbe richiedere più di un anno se condotto da personale specializzato.

Contemporaneamente e per un periodo di tempo non maggiore, l'Unità Operativa CSELT svolgerà la ricerca precedentemente enunciata sulle fibre per fornire valutazioni più sicure sul costo, fattibilità e affidabilità della soluzione in fibra.

- Sviluppo del progetto riguardante l'implementazione del protocollo di accesso EXPRESS con particolare riferimento alla parte del NIM destinata ad implementare tale protocollo.

Sviluppo e definizione dell'Architettura di rete e dei vari livelli di Protocollo.

Anche per lo sviluppo di questi argomenti è necessario l'apporto di competenze sia hardware che software.



### 3.2. Linguaggio di Sistema e Linguaggio Intermedio

Si intende arrivare nel 1981 ad una definizione completa del linguaggio intermedio di Cnet (X-Code). Tale definizione comprenderà oltre alla parte sequenziale tutti gli aspetti di comunicazione inter-node, real-time, fault-tolerance, configurazione. Inizierà dunque la fase prototipale con una prima implementazione di X-Code su S6000 (HOST Olivetti-Cnet) in ambiente emulato ed una successiva su "host" S3000 (nuova linea Olivetti, che si prevede disponibile solo dopo luglio 1982) utilizzando solo una parte minima del nucleo del Sistema Operativo e di "drivers" di Ingresso/Uscita.

### 3.3. Tecniche di Implementazione del Software

Per quanto riguarda le tecniche di implementazione, l'analisi fatta riguarda il linguaggio di programmazione per lo sviluppo del software di Cnet. La scelta, essenziale per rendere la prototipazione del software indipendente dallo sviluppo del hardware del prototipo Cnet, si è concentrata sui linguaggi PASCAL, ADA, e relativi sottoinsiemi. Compito primario del gruppo di lavoro relativo è arrivare alla fine del 1981 con la scelta del linguaggio di implementazione considerando i vantaggi ed i problemi della scelta di Pascal e di ADA.

La scelta implica considerazioni su disponibilità, reperibilità, affidabilità e potenza dell'ambiente di sviluppo. Occorre inoltre provvedere alla progettazione ad hoc di strumenti di trasformazione per poter portare i programmi sull'architettura target.

### 3.4. Ambiente integrato di sviluppo software

Verrà completata la definizione delle specifiche funzionali per gli strumenti listati di seguito, e per tutti quegli aspetti in cui non è coinvolto il modello di parallelismo e di distribuzione del linguaggio:

- . KAPSE: nucleo del sistema di sviluppo
- . Strumenti che operano sulla rappresentazione dei programmi in sintassi astratta, (editor, analizzatori sintattici e semantici, interprete-debugger).
- . Linguaggio di comandi
- . Strumenti per il debugging simbolico, la valutazione e la misura nell'ambiente operativo (sulla macchina target).

Gli altri strumenti dell'ambiente su target (per esempio, linker-loader) verranno analizzati nell'ambito dei gruppi che si interessano del linguaggio di sistema.

Verranno inoltre definiti i requisiti per gli aspetti del debugging di programmi distribuiti.

Non si è ritenuto affrontabile nell'ambito del progetto lo sviluppo di tutti gli strumenti necessari. In particolare, non verrà sviluppato nell'ambito di Cnet il compilatore, che si cercherà di acquisire all'esterno. Una via che si intende verificare è la possibilità di ottenere un compilatore sperimentale, per composizione di alcuni degli strumenti elencati precedentemente e di un generatore di codice intermedio che potrebbe essere sviluppato con metodologie di tipo formale, simili a quelle sopra proposte, dal gruppo del Prof. Bjorner, Politecnico di Lyngby, Danimarca.

## 2.P1.CNET.4. Proposta di ricerca 1982

A differenza di quanto avvenuto nella fase di specifiche funzionali di Cnet, già del 1981 e ancor più a partire dal 1982, l'attività di Cnet verrà focalizzata sugli aspetti prototipali. Infatti, l'attività prevista per il 1982, del piano pluriennale ossia le specifiche di progetto e la prototipazione sono interdipendenti tra loro.

L'attività prototipale non riguarda soltanto i prototipi di Cnet previsti a Pisa e Milano, bensì tiene conto anche della produzione di strumenti che possono essere sviluppati su sistemi diversi e necessari per la produzione software (VAX-UNIX Pisa, VAX-UNIX Milano, VAX-UNIX-OLIVETTI Pisa, VAX-UNIX Torino).

Infatti, i gruppi di lavoro si riorganizzarono intorno a degli obiettivi di progetto che riguardano le cinque linee: Sottorete di Comunicazione, Linguaggio di sistema, Modelli semantici, Ambiente Integrato di sviluppo software e Sistema integrato di automazione degli uffici. L'articolazione dei gruppi di lavoro su queste linee subirà una ulteriore integrazione intorno ai due prototipi Cnet di Pisa e Milano.

### 4.1. Sottorete di comunicazione Cnet

L'attività relativa alla sottorete di comunicazione Cnet è totalmente correlata agli sviluppi già descritti nella relazione sullo stato della ricerca e agli aggiornamenti dei piani 1981. Infatti, le specifiche funzionali della parte trasmissiva della sottorete di comunicazione di Cnet in fibra e la definizione del prototipo sperimentale permetteranno una valutazione finale sulla fattibilità della utilizzazione della fibra come mezzo trasmissivo.

Restano però congelati i parametri previsti per l'architettura della rete Cnet ossia:

- Topologia in grado di connettere stazioni la cui distanza può raggiungere l'ordine di qualche Km. (2 a 10)
- Accesso garantito a un numero di stazioni dell'ordine di qualche centinaio. (256)
- Possibilità di integrare insieme ai dati, anche il traffico vocale e tutti gli altri tipi di traffico previsti nelle applicazioni note come Office Automation. (Videotext)
- Capacità di collegamenti Broadcast.
- Meccanismo di accesso e controllo della sottorete completamente distribuito.

La struttura proposta, consistente in un bus unidirezionale a due parti, (OUTBOUND Channel e INBOUND Channel) nota come EXPRESS-NET (in fase di brevetto tra CNR e Stanford University) può essere realizzabile su diversi mezzi trasmissivi: fibra, cavo coassiale.

Dopo una prima realizzazione prototipale della sottorete in cavo coassiale, con prestazioni degradate rispetto agli obiettivi finali, si intende a seconda dei risultati che scaturiranno dagli studi di fattibilità su l'utilizzazione della fibra, passare ad una ulteriore prototipazione che rispetti i requisiti elaborati. Se i problemi di tipo tecnologico, di costo, affidabilità o manutenzione che la fibra introduce, non possono essere risolti in tempo utile per il progetto si prevede l'utilizzazione del cavo, consapevole che si tratta di sperimentare questo mezzo trasmissivo al massimo delle possibili prestazioni fisiche (34 Mb/sec).

L'attività prevista anche se non esauribile nel 1982 è a sua volta così articolata:

Sottorete di Comunicazione: parte trasmissiva

- Specifiche di progetto ed inizio della fase prototipale della parte trasmissiva EXPRESS-NET realizzata in fibra ottica a 34 Mb/s. Attività supeditata a risultati 1981.
- Specifiche di progetto ed inizio della fase prototipale della parte trasmissiva su cavo coassiale utilizzando la tecnologia esistente, alla velocità di 34 Mb/sec. Attività di ricerca sulla realizzabilità di accoppiatori direzionali passivi su cavo.

Sottorete di Comunicazione: interfaccia mezzo trasmissivo-host (NIM, Network Interface Module)

- Specifiche di progetto ed inizio della fase prototipale del NIM, stabilendo le scelte tecnologiche, i criteri di dimensionamento, la risposta davanti a crash, l'affidabilità. Si noti che il progetto del NIM non comprende la realizzazione del "tranceiver", unità per la transcodifica dei segnali da e verso la rete, e quindi è sganciato dalla tecnologia per esso adottato, dipendendo però dalla velocità scelta per la ricetrasmissione (34 Mb/sec).
- Progetto del Software specializzato realizzando EXPRESS e protocolli a più alto livello per svolgere funzioni di ricetrasmissione delle informazioni, della loro memorizzazione e d'interfacciamento con gli host. Detta interfaccia dovrà essere compatibile con la definizione della macchina astratta Cnet di basso livello (X-Code).

Sottorete di comunicazione: misure delle prestazioni sul prototipo (Spy Monitor)

- . Progettazione del modulo interfaccia dello Spy con la sottorete, con filtro sul riconoscimento del campo destinatario, la specifica del modulo di interfaccia con la sottorete sarà realizzato congiuntamente alla specifica dell'architettura HW e SW del NIM. La realizzazione dovrebbe essere quasi immediata a partire dalla corrispondente piastra del generico NIM.
- . Specifica della piastra di logica di confronto/riconoscimento dello Spy e sua realizzazione, il punto più critico.

- . Progetto del modulo di interfaccia con la macchina host per la memorizzazione on-line dei dati raccolti dallo Spy Monitor, poiché la macchina host che svolge "data reduction" sarà uno degli host di Cnet non ci saranno particolari attività extra per realizzare il modulo di interfaccia tra logica dello Spy e host. Il trasferimento avverrà nel rispetto dell'interfaccia tra sottorete e macchina astratta di Cnet.
- . Specifica e realizzazione del software di "data reduction" sull'host per la funzione di elaborazione statistica e la produzione di reports.

#### Sottorete di Comunicazione: Strumenti di progetto off-line

Sviluppo di un programma di simulazione della sottorete di comunicazione di CNET che modelli il funzionamento del mezzo di trasmissione, dei ricetrasmittenti, del protocollo di accesso al canale, del NIM e delle apparecchiature collegate a quest'ultimo.

Il protocollo considerato sarà principalmente quello proposto da Borgonovo, Fratta e Tobagi (EXPRESS-NET), ma si confronteranno i risultati ottenuti con quelli forniti dalla variante proposta da Ajmone ed Alberengo (MAP).

Le apparecchiature collegate alla rete saranno modellate utilizzando opportuni modelli analitici dai quali saranno derivate le caratteristiche del traffico che la rete deve smaltire.

Nel programma di simulazione verrà usato un livello di dettaglio tale da evidenziare le caratteristiche della sottorete che si riterranno più importanti, introducendo le opportune semplificazioni per contenere i tempi di esecuzione entro limiti accettabili.

Sarà studiata la possibilità di sviluppare un simulatore ibrido, sostituendo modelli analitici appropriati alle corrispondenti parti del modello di simulazione. In ogni caso ciò risulti possibile, il modello ibrido sarà opportunamente validato.

Il risultato che ci si prefigge con lo sviluppo di questo strumento è di poter analizzare in sede di progetto l'influenza che hanno alcuni parametri della realizzazione della sottorete. Esempio tipico di un dato di progetto che può essere dimensionato in questo modo è la dimensione della memoria per l'immagazzinamento dei pacchetti nel NIM.

#### 4.2. Linguaggio di sistema

Per quanto riguarda la macchina intermedia Cnet o X-Code si prevede:

- . Implementazione di X-Code su host Cnet, presumibilmente S-3000, utilizzando solo una parte minima di nucleo e i "drivers" fisici delle diverse periferiche presenti nella configurazione.
- . Bootstrap su S3000 delle primitive di livello superiore realizzate durante l'attività 1981, su S6000.

- . Collaudo delle primitive di comunicazione e configurazione in ambiente fisicamente distribuito, nell'ipotesi di disponibilità del sottosistema di comunicazione in versione prototipale.
- . Estensione di X-Code con i meccanismi primitivi necessari per il "crash-recovery", (backward-error recovery).
- . Estensione di X-Code con i meccanismi di astrazione del controllo per implementare meccanismi di "recovery" per errori software non previsti.
- . Definizione e inizio implementazione dei primi livelli di sistema operativo distribuito.
- . Estensione di X-Code ad una architettura multi-processor nella eventualità che si possa disporre di host MARA-SELENIA.

Le attività precedentemente elencate ed i risultati dell'attività prevista per il 1981 sulle tecniche di Implementazione costituiscono il requisito essenziale per tutte le componenti implementative del Software Cnet.

Sulla base della definizione dell'architettura, si esamineranno le estensioni ADA necessarie per coprire le esigenze di Cnet. In prima istanza si cercherà di inserire estensioni realizzabili attraverso i meccanismi di astrazione offerti da ADA (packages, tipi astratti) in modo da non alterare ADA. Le estensioni di ADA riguardano comunicazione inter-nodo, "fault-tolerance", Configurazione, Real-Time e primitive necessarie per l'implementazione dell'ambiente di sviluppo software. In alcuni casi, probabilmente, occorrerà proporre delle estensioni effettive.

Per quanto riguarda il "backward error recovery" per la gestione dei malfunzionamenti che rendono non disponibili risorse fisiche del sistema (crashes) ed il "recovery" per errori software non previsti è necessario provvedere ad una opportuna estensione di ADA onde incorporarvi i meccanismi primitivi già definiti in X-Code, possibilmente mediante la definizione di un certo numero di packages specializzati.

Gli studi in corso sulle modifiche da apportare ad ADA quando ci sono vincoli di tempo reale stretto, presuppongono anche estensioni sintattiche. Altre estensioni sono previste per trattare la problematica che riguarda la configurazione dei sistemi distribuiti e cioè: la specifica della rete sia fisica che virtuale; la specifica del nodo virtuale; l'allocazione del nodo virtuale sul nodo fisico ed il legame fra nomi sulla rete.

Questi due ultimi aspetti sono necessari per consentire configurazione dinamica e dunque dovranno far riferimento a meccanismi elementari proposti da X-Code.

#### 4.3. Modelli semantici

L'attività si articolerà su due filoni:

- . Produzione delle specifiche semantiche del prototipo Cnet.

Si tratta di un obiettivo strettamente finalizzato al progetto, che sarà raggiunto a fine progetto, producendo: una descrizione formale delle parti innovative dell'architettura (software) Cnet e metodi di prove di correttezza per il software Cnet.

- . Supporto nella ricerca di metodologie da applicare nella produzione delle specifiche semantiche, analisi e proposte di soluzione per i problemi di modellizzazione matematica di queste metodologie e collaudo su sistemi opportuni.

In particolare si prevede:

- . Sviluppo della proposta MHA e derivati, collaudo su ulteriori linguaggi, ed applicazioni a parti di Cnet (primi modelli relativi ad un singolo nodo e modello del sistema inter-node).
- . Elaborazione dei modelli semantici necessari per rendere consistente la descrizione della semantica con particolare riguardo a problemi di sincronizzazione e real time.
- . Analisi di metodologie per le prove di correttezza; per il prossimo anno si prenderanno in considerazione gli sviluppi recenti di sistemi Hoare-like, collegandoli alle caratteristiche del software Cnet.

#### 4.4. Ambiente integrato di sviluppo software

L'attività '82 prevede la prosecuzione sull'attività di ricerca sugli strumenti dell'ambiente su macchina ospite e sul debugging su target e l'inizio dell'attività di prototipazione.

Per quanto riguarda gli strumenti su macchina ospite che agiscono o sulla sintassi astratta, verranno studiati quegli aspetti che sono influenzati dal modello di concorrenza e distribuzione del linguaggio. In particolare, si considereranno le estensioni necessarie per trattare tali costrutti nel linker ad alto livello e nell'interprete. Dato che gli strumenti di questo gruppo sono ottenuti con metodologie basate sulla definizione formale, sarà necessario estendere il metalinguaggio per definire adeguatamente la semantica di tali costrutti.

Per quanto riguarda il KAPSE, saranno disponibili entro il Dicembre 1982, le specifiche di progetto, con particolare attenzione a quegli aspetti che dovrebbero costituire un'estensione sul nucleo definito nel progetto ADA-CEE (che verrebbe reso disponibile a Cnet). Si tratta in particolare, di verificare la possibilità di definire meccanismi che permettano di realizzare, in modo uniforme

- primitive che garantiscono l'open-endedness dell'ambiente
- primitive che permettono di trasformare una struttura dati in un programma eseguibile
- primitive che permettono di definire dinamicamente (possibilmente anche su target) il legame fra la specifica di un package ed il suo corpo.

Il secondo aspetto di interesse particolare nell'ambito del KAPSE riguarda la realizzazione della base di dati di progetto, attraverso una base di dati distribuita. Tale scelta influenza evidentemente le primitive di basso livello che verranno definite nel nucleo.

Il terzo filone di ricerca riguarda il debugging su macchina target. Entro la fine del 1982 saranno disponibili le specifiche di progetto del debugger, ivi compresi gli aspetti relativi a target distribuito. Si noti come tale strumento dipenda in maniera seria da scelte che vengono fatte da chi realizza il compilatore e che non sono quindi, sotto diretto controllo del progetto. Si cercherà quindi di definire un ambiente di debugging su target che imponga il minimo di condizioni nel compilatore e che possa invece essere eventualmente utilizzato appieno se collegato all'ambiente in macchina ospite.

L'attività prototipale prevede nel 1982 l'inizio della realizzazione dei seguenti strumenti:

- a)
  - a1 sistema per la manipolazione di alberi (programmi in sintassi astratta), compresa la gestione della memoria;
  - a2 generatore della rappresentazione interna, guidato dalla definizione formale sintattica, e sua utilizzazione per ottenere un syntax-driver editor per ADA;
  - a3 un parser guidato dalla sintassi;
  - a4 type-checker, controllore delle interfacce, linker ad alto livello, definiti con stile denotazionale;
  - a5 interprete denotazionale di ADA sequenziale;
  - a6 primitive generali che definiscono il sottoambiente di debugging interattivo.

Obiettivo di questo gruppo di strumenti è di realizzare, oltre che una parte dell'ambiente per ADA, un metaambiente adattabile con relativa fattibilità a linguaggi diversi. Si prevede che l'attività di realizzazione si protrarrà per due anni.

- b) Debugger sequenziale su macchina target.
- c) Realizzazione di un nucleo con caratteristiche innovative, rispetto a quelle del progetto ADA-CEE, per quanto riguarda i meccanismi di innovazione e comunicazione (binding dinamico).

L'attività di realizzazione di b) e c) dovrebbe estendersi anch'essa sull'arco di due anni, includendo anche la realizzazione degli aspetti distribuiti.

#### 4.5. Strumenti per l'automazione di uffici

L'architettura lascamente connessa e funzionalmente distribuita di Cnet si presta particolarmente, in linea di principio, come base per un sistema informativo di ufficio (SIU) che supporti le attività di una comunità di utilizzatori.

Di un tale sistema ci interessa mettere in evidenza i seguenti aspetti:

- . Disponibilità di una gamma di strumenti per lo svolgimento di compiti specifici dell'individuo (word-processing, information-management, form-management, etc.) con caratteristiche di: facilità d'uso, alta interattività, integrazione.
- . Funzionalità di comunicazione fra gli utilizzatori, di distribuzione dell'informazione, di definizione, attuazione e controllo del flusso di documenti fra i componenti della comunità.
- . Alta affidabilità, robustezza e persistenza dei dati.

Per quanto riguarda il primo aspetto, esso può essere adeguatamente trattato adottando, per i nodi "work stations", degli elaboratori personali di elevate prestazioni con interazione di tipo grafico.

Gli altri aspetti sono più specifici di un'architettura distribuita e si ricollegano a tematiche, quali lo studio delle primitive di base per Inter Process Communication o dei meccanismi di fault-tolerance, affrontati da altri gruppi di lavoro all'interno dell'obiettivo Cnet.

Per affrontare questi problemi nell'ottica dei sistemi per ufficio, si propone di progettare e sviluppare un prototipo di applicazione che attraverso "verticalmente" tutti gli aspetti, da quelli di sistema a quelli dell'utilizzatore finale.

Il progetto si articolerà in tre parti: realizzazione di spazi di lavoro persistenti (SLP); di un sistema per il trattamento di documenti attivi (STDA); di strumenti per l'automazione di procedure amministrative (SAPA).

Gli SLP garantiscono che lo spazio di lavoro di ciascun utilizzatore, ossia la collezione delle informazioni ad esso accessibili, è in grado di sopravvivere senza danni nè perdite ai crolli del sistema.

Tale caratteristica è essenziale affinché il sistema elettronico possa costituire un'alternativa realistica ai sistemi informativi oggi utilizzerà negli uffici e che, essendo sostanzialmente basati su carta, presentano altissima affidabilità, disponibilità e persistenza.

Un documento attivo è un documento i valori dei cui campi sono legati da relazioni o procedure con altri campi dello stesso o di altri documenti. Intervendendo sul documento vengono innescate azioni per mantenere i vincoli o causare gli effetti specificati dalle procedure.

Il STDA consente:

- . Preparazione interattiva di documenti;
- . Interfaccia con la base di dati dell'ufficio;
- . Elaborazione di bilanci, consuntivi, analisi di previsione;
- . Comunicazione tra workstations trasferendo documenti tramite rete;



. Stampa su formulari predisposti e secondo formati prescelti.

Gli SAPA dovranno invece permettere ad una comunità di utilizzatori non specialisti di automatizzare (anche solo parzialmente) le procedure amministrative che essi stessi mettono in opera per realizzare gli obiettivi posti alla comunità.

Il sistema informativo deve essere utilizzabile per diminuire l'overhead amministrativo, in primo luogo sollevando l'utilizzatore dalle operazioni ripetitive e prive di contenuto professionale che inevitabilmente esistono nelle procedure amministrative e che anzi possono essere enfatizzate dalla introduzione di sottosistemi specifici.

Poiché le procedure amministrative sono tipicamente non ben strutturate, basate su eventi e ricche di situazioni eccezionali non prevedibili a priori istituzionalmente suddivise su varie fasi che devono coinvolgere persone e uffici diversi,

il SAPA dovrà:

- . essere orientato alla automazione parziale delle procedure privilegiando l'intervento dell'utilizzatore per fasi a maggiore contenuto decisionale (tra le quali essenziali sono i trattamenti delle situazioni eccezionali).
- . Permettere la definizione e la cattura di eventi di interesse per gli utilizzatori.
- . Capitalizzare sulla cooperazione fra utilizzatori e comunità quale strumento primario per il raggiungimento degli obiettivi della comunità e per il miglioramento della sua efficienza.

## 2.P1.CNET.5. Suddivisione della ricerca tra le Unità Operative

Unità Operative	Mezzo Trasmissivo	Protocolli SRC	NIM	Strumenti di Misura SRC	Strumenti di Progetto OFF - LINE - SRC	X-CODE	ADA - Distribuito	Modelli Semantici	Fault Tolerance SW	Kapse e Debugging Distribuiti	Strumenti Statici AISS	Applicazione: OAS	Prototipi e Tecniche di Implementazione
CSISEI - MI						*	*	*					*
CNUCE - PI			*								*		
IEI - PI						*	*		*				
IAC - RO								*					
IST. EL. MI	*	*	*										
DIP.SIS. CS							*		*				
IST. EL. PI							*		*				
IST. EL. TO		*			*								
IST.MAT. GE								*					
IST.MAT. RO								*					
ISI - PI							*	*		*	*	*	*
ISI - TO								*			*	*	
OLIVETTI				*	*					*		*	*
SELENIA			*										*
ITALTEL									*		*		*
SYNTAX						*				*			*
S & M											*		*

## 2.P1.CNET.6. UNITA' OPERATIVE

## UNIVERSITA' E ORGANI DEL CNR

- Centro di Studio sull'Ingegneria dei Sistemi per l'Elaborazione dell'Informazione, Milano  
Resp. della ricerca: F. Tisato.
- CNUCE - Istituto del CNR, Pisa  
Resp. della ricerca: F. Tarini.
- Istituto di Elaborazione dell'Informazione-CNR, Pisa  
Resp. della ricerca: M. Fusani.
- Istituto di Elettronica e Telecomunicazioni - Facoltà di Ingegneria, Pisa  
Resp. della ricerca: P. Ancilotti.
- Istituto di Elettronica e Telecomunicazioni - Facoltà di Ingegneria, Torino  
Resp. della ricerca: M. Ajmone Marsan.
- Istituto di Elettrotecnica ed Elettronica - Politecnico, Milano  
Resp. della ricerca: L. Fratta.
- Istituto per le Applicazioni del Calcolo "M. Picone"-CNR, Roma  
Resp. della ricerca: M. Venturini Zilli.
- Istituto di Matematica - Facoltà di Scienze M.F.N., Genova  
Resp. della ricerca: E. Astesiano.
- Istituto di Matematica "G. Castelnuovo", Roma  
Resp. della ricerca: C. Bohm.
- Istituto di Scienze dell'Informazione, Pisa  
Resp. della ricerca: C. Montangero.
- Istituto di Scienze dell'Informazione, Torino  
Resp. della ricerca: M. Dezani.
- Università della Calabria - Dipartimento di Sistemi, Cosenza  
Resp. della ricerca: M. Di Santo.

## INDUSTRIE

- CSELT Centro Studi e Laboratori Telecomunicazioni S.p.A., Torino  
Resp. della ricerca: A. Luvison.
- ITALTEL S.p.A., Settimo Milanese  
Resp. della ricerca: S. Belli.
- Ing. C. OLIVETTI & C. S.p.A., Pisa  
Resp. della ricerca: D. Monaco.
- SELENIA S.p.A., Roma  
Resp. della ricerca: S. Bottalico.
- SYNTAX S.p.A., Milano  
Resp. della ricerca: A. Fedeli.
- SYSTEMS & MANAGEMENT S.p.A., Pisa  
Resp. della ricerca: E. Dameri.
- ISED/CONTRAVES S.p.A., Roma  
Resp. della ricerca: R. Giannetti.

2.P1.CNET.7. PROSPETTO DELLE RICHIESTE FINANZIARIE**SPESE DI INVESTIMENTO**

Richiesta di finanziamento relativo all'acquisto di apparecchiature indispensabili per l'attività prototipale sia a Pisa che a Milano. Host adatti per applicazioni di automazione ufficio e ASS. NIM per collegamento degli Host alla rete.

a totale carico Industria	<u>L. 304.000.000</u>
a totale carico CNR	<u>L. 60.000.000</u>

**SPESE DI FUNZIONAMENTO**

Spese per materiale di consumo e gestione e manutenzione indotte dalle spese di investimento e prestazioni tecniche, affitto linee, collegamenti in Rete, da dividere tra le due sedi che ospiteranno i prototipi Cnet.

L. 357.100.000

Spese di funzionamento per le Unità Operative presso Enti Pubblici relative a 22.1 Ric. Equiv. per:

- Documentazione
- Viaggi e missioni
- Professori Visitatori
- Stampa materiale
- Uso di calcolatore

Voce B e C	<u>L. 267.900.000</u>
------------	-----------------------

Spese di funzionamento per le Unità Operative presso Società private o a partecipazione statale relative a:

- N. Ore ricercatore o equivalente valutate alle tariffe orarie di L. 43.750 Anno Ricerc. Equiv. e L. 37.500 Anno Ricerc. Equiv. 34.3
- Stampa di materiale
- Utilizzazione laboratorio

Totale 100%	<u>L. 2.949.500.000</u>
di cui a carico CNR	<u>L. 2.184.081.000</u>

<b>Totale Finanziamento per il 1982</b>	<u>L. 3.938.500.000</u>
---	-------------------------

<b>Totale Finanziamento a carico Industria</b>	<u>L. 1.069.419.000</u>
--	-------------------------

<b>Totale Finanziamento a carico CNR</b>	<u>L. 2.869.081.000</u>
--	-------------------------

Finito di stampare nell'ottobre 1982  
dall'ETS Pisa

33. R. Marcogliese, R. Novarese, The Cnet Monitoring Center and its Performance Monitoring Function
34. G.F. Rossi, Meccanismi di comunicazione tra processi concorrenti nel linguaggio Ada
35. C. Ghezzi, D. Mandrioli, F. Tisato, R. Zicari, A Critical Analysis of the Ada Task System with Respect to Real Time Programming
36. N. Cocco, C. Ghezzi, D. Mandrioli, F. Tisato, Mechanisms for Building Abstract Machines
37. G.F. Rossi, Bibliografia sul linguaggio Ada (per argomenti, parzialmente commentata)
38. C. Ghezzi, D. Mandrioli, F. Tisato, A Machine Architecture Supporting Asynchronous Activities
39. M. Coppo, S. Ronchi Della Rocca, Analisi di alcuni modelli di sistemi distribuiti
40. M. Venturini Zilli, Nondeterminism and Computability
41. P. Ancilotti, M. Fusani, Azioni atomiche: strumenti e proprietà
42. P. Ancilotti, Relazione sull'attività del gruppo di lavoro su Fault-Tolerance
43. I. Castellani, P. Franceschi, U. Montanari, Labeled Event Structures: a Model for Observable Concurrency
44. M. Bellia, P. Degano, The Tool of AISS for Program Editing, Abstract Representation Construction and Program Displaying
45. R. Barbuti, A. Martelli, Execution and Debugging Tools in AISS Based on Denotational Semantics Concepts
46. E. Dameri, C. Simonelli, Rappresentazione intermedia di programmi in AISS
47. G. Attardi, Office Information Systems design and Implementation
48. R. Barbuti, M. Bellia, A. Martelli, E. Dameri, C. Simonelli, P. Degano, G. Levi, Towards the Derivation of an Experimental Programming Environment from Language Formal Specifications
49. P. Amatruda, Considerazioni sulle eccezioni e loro gestione: conseguenze sui requisiti di un linguaggio di programmazione
50. F. Bruni, Sull'inadeguatezza di qualsiasi insieme finito di strutture per rappresentare reti a livelli di astrazione successivi
51. C. Dionisio, R. Vannini, Survey sul trattamento digitale del segnale vocale
52. F. Giannessi, Tool Composition and Interprogram Communication Mechanisms
53. E. Lazzari, F. Nicodemi, Proposta per il Debugger
54. F. Gallo, Kapse Database Functional Specification
55. F. Gallo, Kapse Database Overview
56. P. Inverardi, G. Vallario, Alcune considerazioni sul KAPSE
57. M. Ajmone Marsan, G. Albertengo, Map: an Insertion Protocol for an Unidirectional Bus Local Network
58. M. Martelli, F. Tarini, Primitive di comunicazione
59. F. Borgonovo, L. Fratta, F.A. Tobagi, The Express Net: A Local Area Communication Network Integrating Voice and Data
60. P. Inverardi, G.N. Vallario, KAPSE: Un costrutto limitato di acquisizione e riconfigurazione dinamica per ADA
61. E. Astesiano, G. Reggio, E. Zucca, Operational Frameworks for Semantic Description of Concurrent Languages. With an application to ADA-like languages
62. M. Di Santo, L. Nigro, W. Russo, Impiego dei regimi di controllo per Forward e Backward Error Recovery

63. A. Pistorello, Aspetti di configurabilità di un sistema di elaborazione in un ambiente distribuito
64. R. Martucci, Introduzione al linguaggio CHILL
65. S. Belli, R. Impagnatiello, Elementi di Fault Tolerance nei sistemi di elaborazione e nelle loro applicazioni nel campo delle telecomunicazioni
66. P. Colombo, C. Ghezzi, D. Mandrioli, A. Tecchio, F. Tisato, R. Zicari, X-CODE
67. R. Barbuti, M. Bellia, P. Degano, G. Levi, A. Martelli, Programming Environments: Deriving Language Dependent Tools From Structured Denotational Semantics
68. P. Inverardi, U. Montanari, G.N. Vallario, How to program an APSE (almost) completely in ADA
69. F. Giannessi, F. Nicodemi, Architettura e funzionalità del sistema di Debugging di programmi ADA
70. P. Inverardi, G.N. Vallario, Architettura di ambiente e struttura del KAPSE
71. M. Chesi, E. Dameri, P. Franceschi, G. Mainetto, C. Simonelli, Specifiche e architettura di un sistema per la manipolazione di alberi di sintassi astratta in un ambiente integrato di sviluppo software.
72. M. Chesi, P. Franceschi, Il Pascal e gli strumenti di programmazione di UNIX(TM) per la realizzazione del prototipo AISS