

# DOCUMENT REVISION HISTORY

Doc. Revision	Date	Changes
1	June 24, 1992	Pre-release copy

## Contents

1	Scope	5
2	Related Documents	5
3	Glossary Of Terms	5
4	GCT Application Model	5
4.1	Top Level Application Entities	6
4.2	Blocks	6
4.2.1	Block Interface	7
4.2.2	Block Internals	7
4.2.3	Block Body	7
4.2.4	Block Types	7
4.3	Block Hierarchy	8
4.4	Libraries	10
4.5	Projects	10
5	GCT Architecture	10
5.1	Open Architecture	10
5.1.1	Use Of CDL	10
5.1.2	CDL Object Manager	14
5.1.3	Incorporating Additional Front-End Utilities	14
5.1.4	Incorporating Additional Target Support	14
5.2	Multiple Platforms	15
5.3	Construction	15

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

2

5	General User Interface Features	16
5.1	Introduction To Menu Bar Usage	16
7	GCT Views	17
7.1	Philosophy of Operation	17
7.2	Main Views	19
7.2.1	Initial Banner	19
7.2.2	Main Menu	19
7.2.3	Project View	22
7.2.4	Library View	24
7.2.5	Configuration View	24
7.2.6	FMC Target Support	24
7.3	Independent Views	24
7.3.1	Block View	24
7.3.2	Library Browser	29
7.3.3	Trace	29
8	SFC Editor	29
9	ST Editor	31
10	Future Utilities	31
10.1	Hierarchical Browser	31
10.2	User Screen Definition	32
10.3	Diagnostics	32
10.4	Simulation and Debug	32
10.5	Source and Version Control	32
10.6	Additional Target Support	32
10.7	FBD Editor	32
10.8	Other Editors	33

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

3

11 GCT Topics	33
11.1 Help	33
11.2 Cut And Paste	33
11.3 Application Documentation	33
11.4 Run-Time Facilities	33

## List of Figures

1 Relationship of Entities within a Configuration	6
2 Hierarchy Of Blocks Within An Application	9
3 GCT Project	11
4 High-Level GCT Dataflow Diagram	12
5 GCT Dataflow Diagram showing Target Independence	13
6 Relationships between GCT Views	18
7 Main Project View	23

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER NO NUMBER, For Information Only	SHT. 4

# 1 Scope

This document is the functional specification of a Group Configuration Tool (GCT), a software tool providing a highly interactive, multi-windowing, graphical interface for the creation and management of control applications. GCT is an evolution from the specification of the Generic Application Programming Environment (GAPE).

This specification gives an introduction to the control model supported by GCT, outlines some architectural considerations and provides an overview of how a user will interact with the system.

An initial implementation of GCT is planned to coincide with the product release of the Fast Machinery Controller (FMC). This initial release of GCT will include a subset of the features outlined in this specification (restrictions are noted in the text).

## 2 Related Documents

- [1] IEC DIS 1131-3 (Programming Languages for Programmable Controllers)
- [2] CDL Specification (NO NUMBER), J.W. Juer
- [3] HP024105: Resource Manager Release 1 Specification, J.W. Juer
- [4] IEC65A Control Languages - A Practical View, J. Juer and I.P. Hughes
- [5] GAPE Preliminary Specification, M.S. Dillamore and J.J. Oliver
- [6] An Overview Of The Resource, I.P. Hughes

## 3 Glossary Of Terms

- CDL - Configuration Definition Language used to specify all aspects of a distributed control strategy.
- FBD - Function Block Diagram graphical representation of continuous control strategy.
- POU - Program Organisation Unit is a service, program or function block.
- SFC - Sequential Function Chart graphical representation of sequential control strategy.
- ST - Structured Text textual language to specify all run-time aspects of control strategy.

## 4 GCT Application Model

GCT enables the user to construct control applications based on the methodology defined in [1] and developed in [4].

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	5

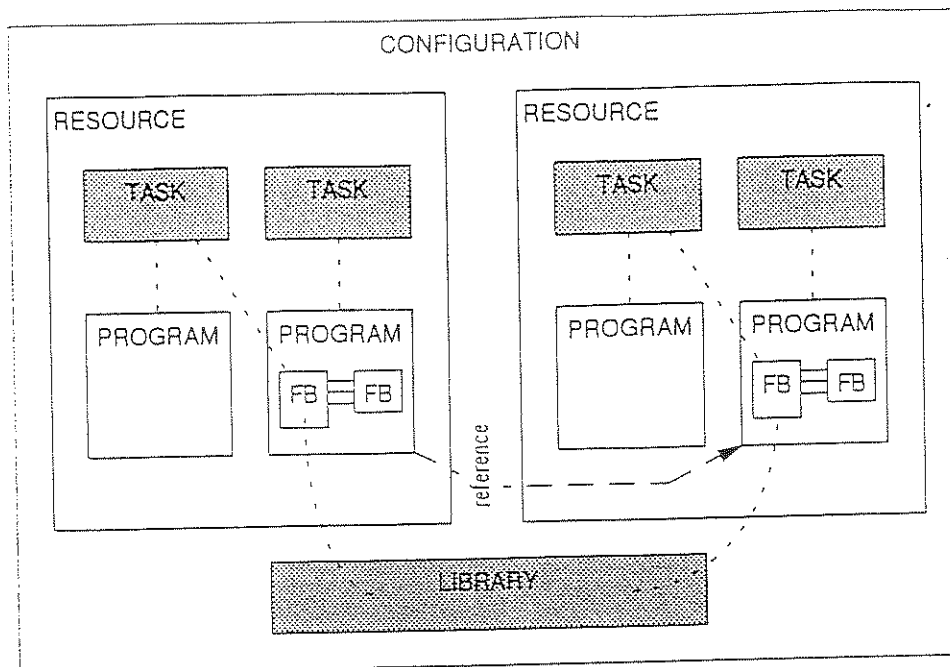


Figure 1: Relationship of Entities within a Configuration

#### 4.1 Top Level Application Entities

The highest level application entity is known, in IEC terms, as a **configuration**. A configuration consists of one or more resources which will be networked together. A resource corresponds to a largely self-contained physical unit within the configuration, such as a Eurotherm PC3000 system with a single LCM (Local Controller Module).

The Resource level of CDL consists of a set of blocks that can be wired together, and divided into subsets that run under the control of a task.

A resource definition can be modified on line. This is desirable to debug parts of an application when other parts of a control system need to be kept running. This concept will be supported by allowing dynamic on-line rewiring of blocks, deletion and addition of block and task instances at the Resource level of the IEC 1131-3 hierarchy. GCT will (in later releases) support this.

Resources have the ability to reference blocks and parameters in other resources, with bindings between resources being resolved at run-time. This late binding facility is considerably more flexible than the basic IEC model, allowing e.g. a new PO resource to access values from a PC3000 resource which had been previously defined, and without modifying the PC3000 resource.

#### 4.2 Blocks

All application entities within this methodology are blocks which have an interface, internals and a body.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool  Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER NO NUMBER, For Information Only	SHT.  6

### 4.2.1 Block Interface

The block interface consists of a number of parameters that may be viewed and used by other blocks. The interface is independent from the implementation of the body of the block and allows a common view of all blocks.

### 4.2.2 Block Internals

The internals of a block specify the entities that are referenced in the implementation of the body of the block. The representation of the internal declarations is also common to all block types and is independent from the implementation of the body. Some internals are implicitly declared eg an SFC body includes step, transitions and actions.

### 4.2.3 Block Body

The body of a block is the actual algorithm that is associated with the block. The body may be implemented using one of a suite of IEC languages (eg SFC, ST, etc), or using another language (eg C, spreadsheet, etc).

The representation of the body of a block will depend on how it was originally defined. When the body of a new block is first opened for edit, an option is given to select one of a number of editors that are appropriate for the given block type. The selected editor will then be invoked to allow the body to be defined. Thereafter the body will retain its body editor type and whenever it is selected for display or modification, the respective editor will again be invoked.

In some cases it will be possible to display or transform a block body into a simpler representation (usually to ST text), eg SFC to ST. It will not be possible to transform to a more complicated body type although default generation of graphics may be supported in future releases.

Initially only ST text and SFC body editors will be supported. The details of these are described in later sections.

### 4.2.4 Block Types

Block types supported include:

- **Configuration**  
Specifies one or more networked resources that are allocated to run on physical nodes.
- **Resource**  
Contains resource level blocks of type task, service, program and function block and the wiring between them.
- **Task**  
Used to specify the execution rate of blocks that are associated with it.
- **Program**  
A program is a function block that can only be instantiated at the resource level.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b>  © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	7

- **Function Block**

A function block consists of a number of input parameters that will be processed to generate a number of output parameters. The values of the input and output parameters will be retained from one execution of the block to the next.

- **Service**

A service block is internal to a program or function block. Services are a CDL extension to the IEC65 standard and are designed to provide remote procedure call and synchronisation of tasks across the networks. Services provide extra methods for a program or function block resulting in modular block definitions and making it easier to construct distributed applications.

- **Function**

A function has one or more named input parameters which will be processed to return a value of a specified data type.

- **Step**

A block that is internal to another block that is defined using the SFC graphical language. A step specifies a number of actions and qualifiers that will be executed when the step is active.

- **Transition**

A block that is internal to another block that is defined using the SFC graphical language. A transition specifies a condition which will cause previously active step(s) to be cleared and subsequent step(s) to be made active.

- **Action**

A block that is internal to another block that is defined using the SFC graphical language. An action specifies some code that will be executed when the steps that it is associated with are active.

Figure 1 illustrates the relationships between the different entities within this methodology.

### 4.3 Block Hierarchy

As has been said, one of a number of languages may be used to define the body of a block. Many of the languages involve either the explicit or implicit internal declaration of other blocks.

- A configuration may contain resources.
- A resource may contain task, services, programs and function blocks.
- A service may contain function blocks and may invoke functions. A service body may be defined using SFC.
- A program may contain function blocks and may invoke functions. A program body may be defined using SFC.
- A function block may contain function blocks and may invoke functions. A function block body may be defined using SFC.
- A block with an SFC body (eg service, program, function block, action), will contain steps, transitions and actions.
- An action body may be defined using SFC. Alternatively an action may invoke functions.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	8

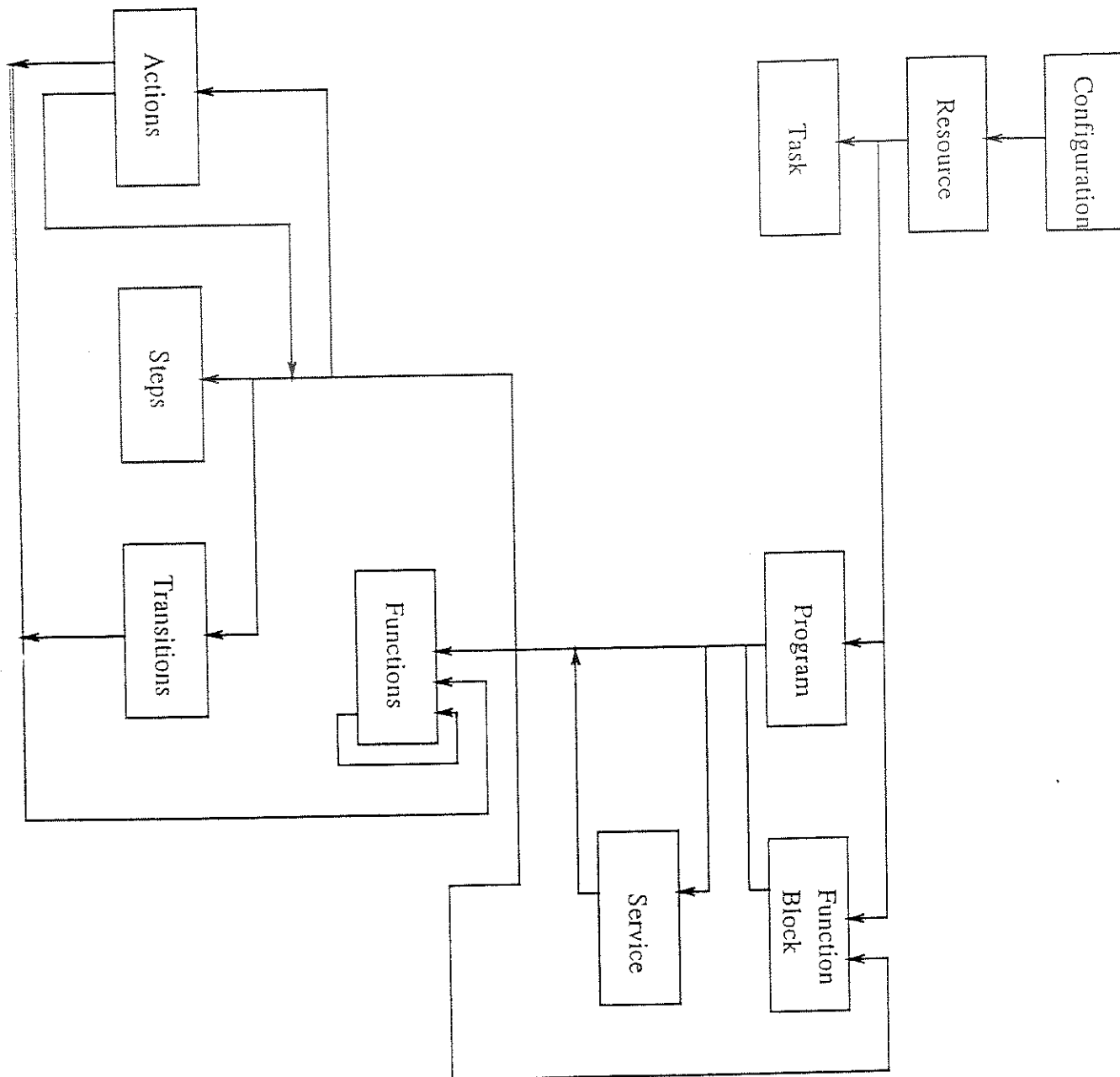


Figure 2: Hierarchy Of Blocks Within An Application

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

DOCUMENT  
REVISION

1

**EI**

Group Configuration Tool

Provisional Functional Specification

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

9



Thus it can be seen that the application model is hierarchical. Figure 2 shows the hierarchical relationship between the different types of blocks in an application.

GCT being based on this hierarchical model will equally support top-down, bottom-up or middle-out configuration of the control strategy.

## 4.4 Libraries

A benefit of the hierarchical, block model is that applications will be highly structured and will encourage the reuse of previously defined blocks. To this end, GCT incorporates library management features. Blocks may be grouped into libraries based on criteria such as specific run-time target functionality or an area of application. Applications can make use of previously defined libraries. Standard Libraries will be supplied by Eurotherm. GCT may also be used to generate a user library which can in turn be used by other applications.

## 4.5 Projects

A number of configuration definitions may be used to allocate a set of resources to physical nodes on a network in different 'configurations'. A number of alternate resource definitions may be used on the same physical network to effect different control strategies. GCT manages such a set of related configurations and resources in a project.

As GCT is used to define hierarchical resources, new blocks will be created. These blocks will be stored in a library that is private to the current project and will not be freely accessible to other applications. It will however be possible to use library management features to take a block from a project library and include it in a shared library.

Figure 3 illustrates the entities managed within a GCT project.

# 5 GCT Architecture

## 5.1 Open Architecture

The GCT software will consist of a suite of front-end utilities that will act on an underlying database (GCT Object Manager), supporting the creation and management of CDL objects (see Figure 4). The GCT architecture will be highly modular, with the disparate utilities having a minimal interface with the rest of GCT (except via the Object Manager), and such that additional utilities and support for additional target control systems may be easily incorporated as required (see Figure 5).

### 5.1.1 Use Of CDL

CDL provides a canonical, portable, extensible representation of all aspects of a control strategy. The use of this common representation will allow the sharing of tools and of application code between multiple configurers and target systems.

GCT will be able to import free format CDL definitions and present them for display and editing. In the future GCT utilities will also support the generation of default graphics to allow a plain textual definition to be viewed graphically.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool  Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER NO NUMBER, For Information Only	SHT.  10

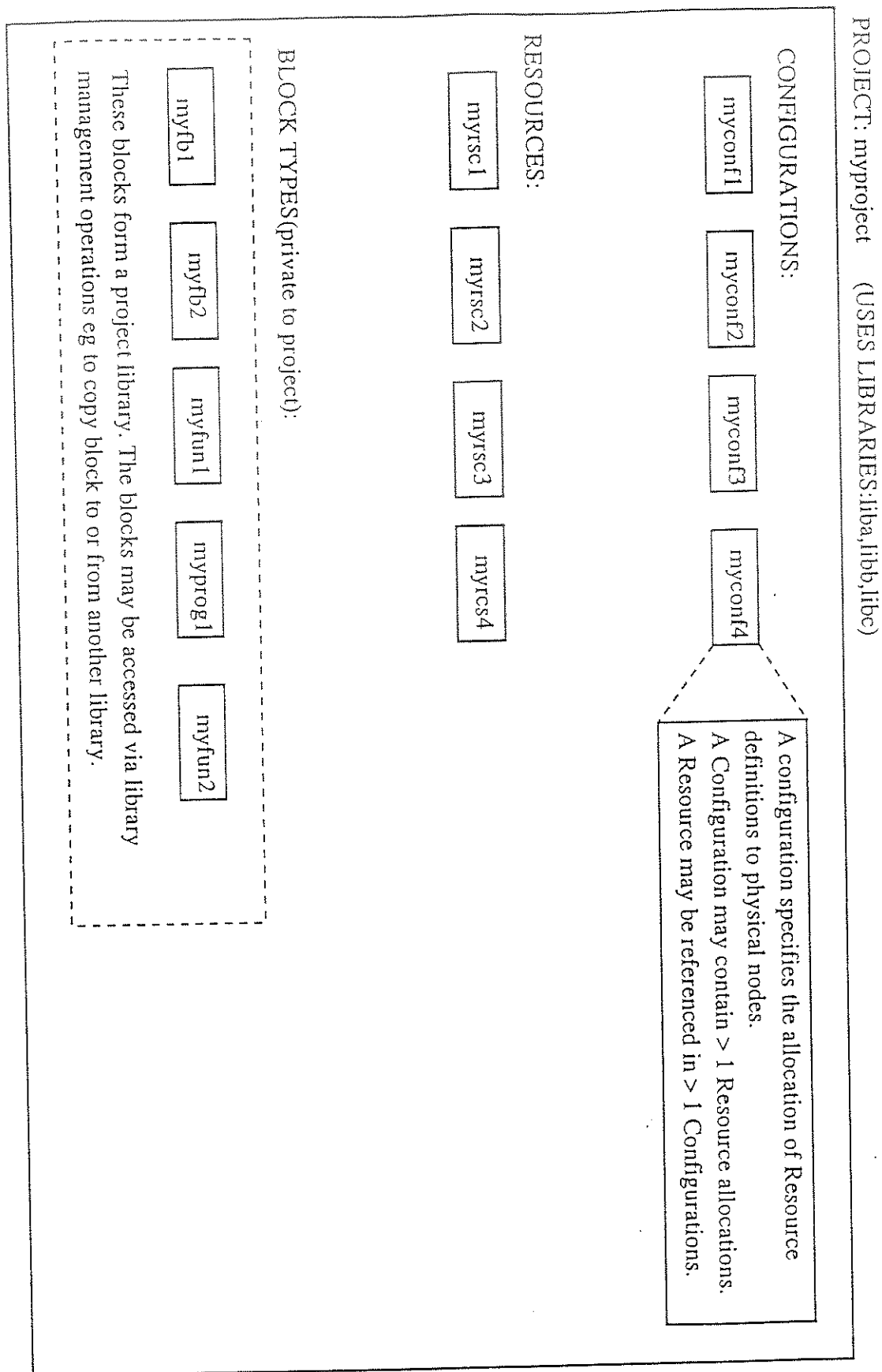


Figure 3: GCT Project

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION	Group Configuration Tool	
	1	Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	11

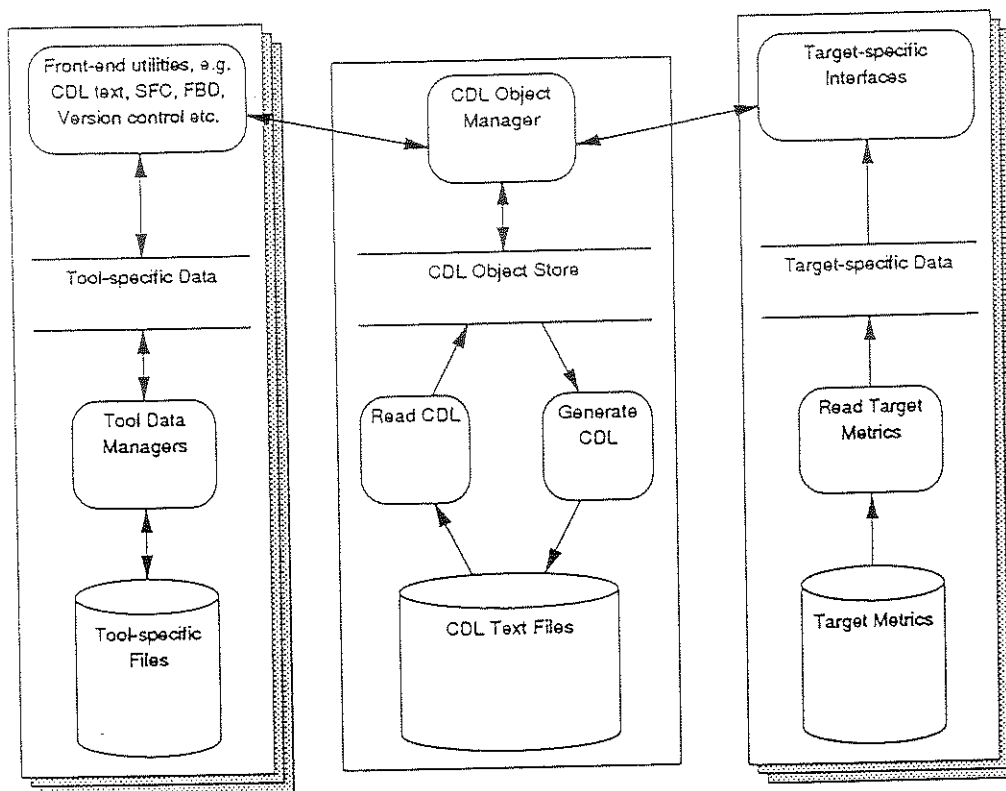


Figure 4: High-Level GCT Dataflow Diagram

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

12

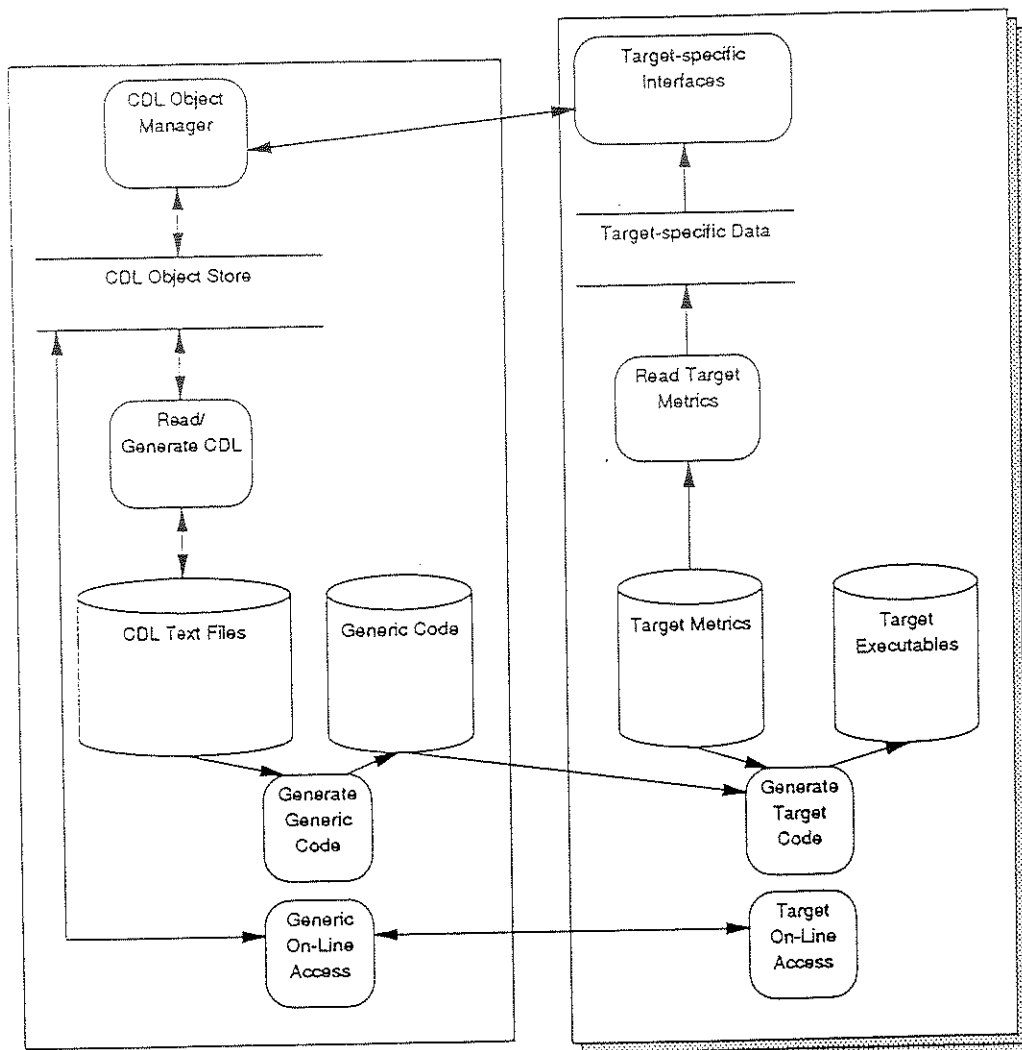


Figure 5: GCT Dataflow Diagram showing Target Independence

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

13

### 5.1.2 CDL Object Manager

The purpose of the CDL Object Manager is to offload all aspects of managing CDL objects from the various front-end utilities that wish to manipulate CDL objects. Functions such as creation, parsing, storing, loading, browsing, code generation, on-line access, download, etc will be handled by the Object Manager in a consistent way and will not have to be reproduced and maintained by multiple applications.

### 5.1.3 Incorporating Additional Front-End Utilities

In the GCT architecture, the GCT Object Manager will support all the aspects of the creation and management of CDL objects. This means that all editors that are capable of manipulating CDL objects may use the Object Manager and will not have to support this management themselves.

An example of this would be the MicroCell spreadsheet representation of an SFC. The spreadsheet representation uses columns to represent steps. Each column has a transition associated with it and the next step(s) to make active. Variables are listed in the leftmost column. If a variable is to be assigned to in a given step, then the value or expression is specified in the row of that step column that corresponds to the variable in the leftmost column that is to be updated. A spreadsheet editor may be incorporated into the GCT by calling Object Manager transactions to create steps, transitions and actions.

Other information required by the spreadsheet editor may be handled by the Object Manager using attributes. CDL supports the definition of user attributes that may be associated with any CDL object. The spreadsheet editor might want to save the row numbers used by the named variables that will reside in the left hand column of the spreadsheet.

The Object Manager would then handle the parsing, storing and code generation for that CDL definition.

If necessary a front-end utility may also maintain files specific to its own format eg for editor characterisation.

### 5.1.4 Incorporating Additional Target Support

The Object Manager generates CDL for the specification of all aspects of a control strategy. CDL is an extensible language and may be used to include user defined attributes. It is this use of a well defined, canonical representation that enables GCT to be a generic configuration environment for multiple targets.

GCT may be configured to allow support of a target system provided that the control model for the target system can be represented by CDL (or a subset of CDL). The target need not support all aspects of CDL, eg dynamic reconfiguration of the resource level. Many run-time systems consist of a pre-defined set of blocks and do not support user defined blocks. They correspond well to the Resource level of CDL. Examples of such systems are T1000 and (possibly) SSD Link.

In order to support a new target the areas to address are:

- Target Code Generation

The GCT Object Manager outputs the definition of a control strategy in CDL text.

A translation utility must be developed to generate target code from the CDL definition. A defined procedure will allow this utility to be installed into the GCT environment and will be invoked by the Object Manager for code generation for the given target.

The Object Manager may also be used to translate CDL code into C that can be compiled and executed on a standard CDL engine. Although the execution engine of a specific target may be able to run this C code, there will most probably be at least a minimal amount of target specific code required to be

UNCONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	14

generated. In some cases all the CDL to target code may be specific to a target, eg if assembler or interpreted tokens are required.

- **Target Code Build**

After target code has been generated, a final build process must be able to generate the final runtime image from a set of target code files eg by linking with target symbols.

This build utility will be installed into the GCT environment in the same way as the code generation utility.

- **Target Comms Driver**

The Object Manager will provide a generic interface for on-line access from GCT utilities to live target systems. Ultimately default comms will be based on the Resource. Specific communications driver must be provided to allow on-line access to live applications for each supported target.

- **Target Specific Front-End Utility**

A target specific front-end utility may be required. This can be incorporated into GCT as described in the previous section.

## 5.2 Multiple Platforms

GCT will be portable across the X window system (initially on Interactive 386/ix) and Microsoft Windows (version 3.0 upwards).

The OSF/Motif library will be used in the construction of the version of GCT for X, and this version will be designed to work best when used in conjunction with the Motif Window Manager.

On PCs, GCT will support the following display standards: EGA, VGA, Super VGA (800 × 600 upwards). Due to the device independent nature of X and MS Windows, support for future display standards of higher resolution (e.g. XGA) will be automatic.

## 5.3 Construction

The GCT development will take advantage of a high-level GUI toolkit to increase productivity and ensure consistency across supported platforms. The toolkit will support interfacing with other third party libraries (e.g. GMS on Unix). It is currently envisaged that the toolkit used will be XVT. Use of XVT would also allow portability to MAC and OS/2.

It will be coded primarily in C++, and will reuse code (either C or C++) from existing programming tools where possible.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool  Provisional Functional Specification	
<b>EUROTHERM</b> <small>© Copyright 1992 Eurotherm Limited</small>	<b>EI</b>	DOCUMENT NUMBER NO NUMBER. For Information Only	SHT.  15

## 6 General User Interface Features

At the highest level GCT will be driven from a horizontal menu bar which will remain at the top of the screen.

All GCT utilities will have a common 'look and feel'. Utilities will run inside windows and use will be made of dialogue boxes, list boxes, icons, and other standard widget types, to facilitate user interaction.

General user interface requirements may be summarised:

**Consistent:** Use of user interface constructs for similar operations will be consistent across all utilities.

**Compliant:** The design and behaviour of the user interface must be in accordance with accepted GUI standards such as Motif and Windows style guidelines.

**Responsive:** The user interface must appear responsive to user action at all times.

**Input devices:** GCT will be driven most efficiently using a mouse, but it will be possible to work with keyboard only.

**Accelerators:** Accelerator keys will be defined for rapid access to frequently used menu options.

**Non-modal:** GCT will contain minimal modality; in general, all utilities will be accessible at all times.

**Multi-window:** The user will be able to view and modify multiple areas of the configuration simultaneously, using different utilities or multiple copies of the same utility as required.

**Interoperable:** It will be possible to exchange data between GCT utilities, or between a GCT utility and third party utilities, using a clipboard and other mechanisms (such as DDE) where appropriate.

**Multi-user:** It will be possible for multiple users to work on applications simultaneously across a network; library management will control locking and updating of the various components of a configuration.

**International:** GCT is targeted at an international market and will therefore provide full multi-language support, also taking into account date and time formats etc.

### 6.1 Introduction To Menu Bar Usage

GCT menu bars can be divided into two categories: the main menu which allows access to operations affecting the entire GCT environment, and menu bars for other views which contain commands for a specific editor (such as deleting a marked section of ST text or printing a particular SFC).

The following description of techniques for accessing menu bar commands applies equally to the main (global) menu bar and to view (local) menu bars.

The structure of a menu bar consists of a horizontal menu bar, whose entries corresponds to a pull-down menu of commands related to a particular area of functionality (e.g. file handling or accessing help).

Items in the menu bar may be accessed using the mouse by clicking on the required item, or using the keyboard by pressing the Alt key in conjunction with the underlined (usually the first) letter of the item (e.g. Alt-M to access the Mode menu).

Having selected a particular pull-down menu, items within the pull-down may be selected either by clicking on the item with the mouse, or by pressing the Alt key in conjunction with the letter of the item which is underlined (e.g. Alt-S for Save or Alt-X for Exit, both of these being in the File pull-down).

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	16

Some regularly used commands from the pull-down menus will have corresponding accelerator keystrokes, so that a single keystroke can activate the command without first having to access a pull-down menu. Where a menu item has a corresponding accelerator key sequence, this sequence will appear alongside (to the right of) the item name in the pull-down menu. At present, this specification does not address the assignment of specific accelerators, although a number of these will be determined by style guidelines.

Menu items which lead to a dialog box (e.g. to allow the setup of command options, or for confirmation) are followed by an ellipsis (...) when displayed.

## 7 GCT Views

The following sections set out the primary views (modes of interaction) in GCT, and the operations supported by each view. The specific details of the user interaction for each operation are not included but an overview of the style of interaction is given as an introduction. Where specific menu structures are given, these are for the purpose of presenting the operations available. The actual contents and presentation of the menus will be subject to change.

Figure 3 (page 18) gives an overview of the relationship between the different views within GCT. A labelled box in the diagram represents a GCT view. The boxes are labelled with the section number in this document that describes that view. An arrow indicates that another view may be opened from that view.

### 7.1 Philosophy of Operation

There are two types of GCT views. The first type are those that are displayed in the main GCT window, and that are controlled by the main menu bar — these are termed *main views*. The second type are displayed in their own windows with their own menu bar these are termed *independent views*. There is only ever one main view active at a time, but there can be multiple active independent views.

GCT operates on the following types of object,

#### The Project

GCT operates within a project context. The project acts as a 'folder' within which Resources, Configurations and the Project Library are stored. The project context maps directly onto a directory. The project context can be changed from within the tool using File, New Project... or File, Open Project....

A Project is represented and managed by the Project main view (see section 7.2.3 and figure 7, page 23)).

#### Resources

The Resource represents the control strategy executing within a node. Resources belonging to a project are listed and managed in the Project main view.

A Resource is represented by a Block independent view by the editor that was used to create the Resource (see section 7.3.1).

#### Configurations

Configurations represent the distribution of Resources across physical networks. In Stage I, GCT is limited to a single node networks so the concept of Configuration definition may not be evident. Configurations belonging to a project are accessed and managed via the Project main view.

A Configuration is represented by the Configuration main view (see section 7.2.5).

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

17



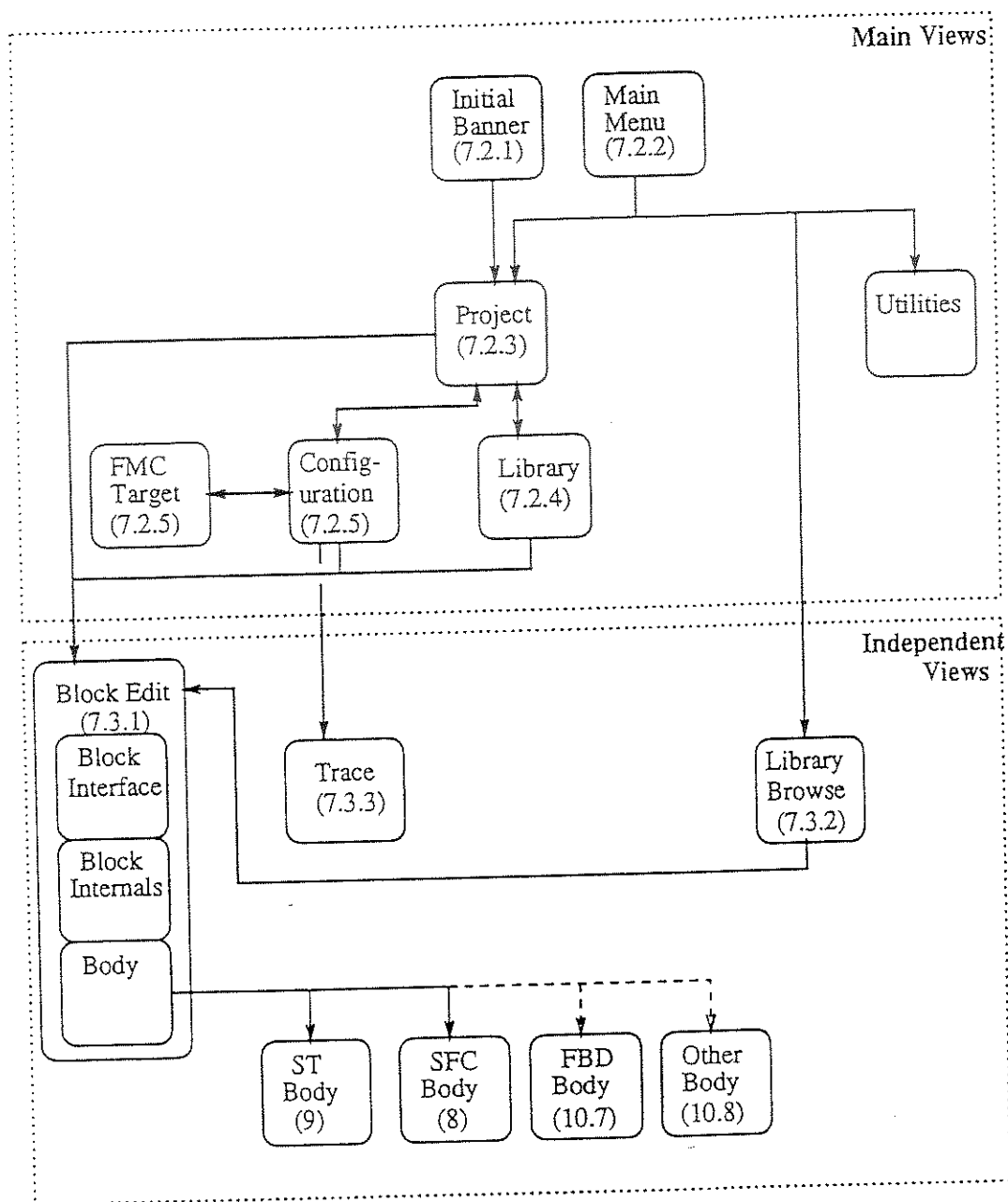


Figure 6: Relationships between GCT Views

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

DOCUMENT  
REVISION

1

**EI**

Group Configuration Tool

Provisional Functional Specification

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

18

## Project Library

The Project Library is a folder containing all user defined block types that are created within a project. Its behaviour and facilities are identical to those of a Shared Library. The Project Library is accessed and managed via the Project main view.

A Project Library is represented by a Library main view (see section 7.2.4) or a Library Browser independent view (see section 7.3.2). Note that when the Library Browser is invoked from the Project main view the main menu bar can be used for library management functions (e.g. copy blocks to another library). In this context the Library Browse behaves as a main view.

## Shared Libraries

A Shared Library is a folder containing user defined block types that may be used by more than one project. Their behaviour and facilities are identical to those of a Project Library. Shared Libraries are accessed and managed via the Project main view.

A Shared Library is represented by a Library ~~Browse~~ independent view (see section ~~7.3.2~~ <sup>7.2.4</sup>).

## Block Types

Block types are created in a Resource or Library context. They are always saved in a library (Project or Shared). Block types are managed from the Library Browse view in main view context.

A block type is represented in a Block independent view by the editor that was used to create it.

When GCT starts up, it must establish a project context. By default the project context is deemed to be the directory that is current when GCT is invoked. The initial project context can also be set by declaring an appropriate environment variable or from within the tool using Utilities, Setup, Defaults....

GCT may be invoked naming a resource, in which case the corresponding block independent view will be opened as well as the Project main view.

## 7.2 Main Views

Main views all occupy the main window. They are controlled from the main menu bar.

### 7.2.1 Initial Banner

The initial view presented on start up is the GCT welcome screen. Besides a welcome message and copyright notification, it (may) contain licensing information. This view is replaced by the Project main view once initialisation is complete.

### 7.2.2 Main Menu

The Main Menu is attached to the main window which contains, in turn, all the main views. The main menu operates on items in the current main view and also provides other functions global to the GCT environment. As well as the main view and the main menu, the main window will also display a small amount of global status information such as the current date and time.

The menu bar window will, by default, remain permanently at the top of the screen while GCT is running. The user will be able to select options from the main menu at any time and with any combination of views on screen, inappropriate menu items being 'greyed out' to indicate their unavailability.

The layout of the items in the menu bar will be as follows:

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	19

Each item of the menu bar represents an entry point to a pull-down menu of associated functions.

Each of the pull-down menus corresponding to top-level menu items will now be described. Menu items which lead to a dialog box (e.g. to allow the setup of command options, or for confirmation) are followed by an ellipsis (...) when displayed.

### File Menu

The File menu provides the means by which typical file-handling operations are performed on the type of object in the current main view. Menu options typically apply to whatever objects are currently selected in the current main view — it is possible to select more than one object currently.

The File pull-down menu contains the following items:

New Project...	greyed out if not on Project View
Open Project...	greyed out if not on Project View
Move...	greyed out if no objects selected in main view
Copy...	greyed out if no objects selected in main view
Delete...	greyed out if no objects selected in main view
Print...	greyed out if no objects selected in main view
Validate	
Build...	
Close	
Exit...	

- File, New Project...

This item leads to a dialog box in which the user specifies the name of the project and the parent directory under which it will be stored. It is greyed out if the main view is not a project, or if there are any independent views active. It resets the Project main view to the new (empty) project's context.

- File, Open Project...

This item leads to a dialog box in which the user specifies the name of the project and the parent directory under which it is stored. It is greyed out if the main view is not a project, or if there are any independent views active. It resets the Project main view to the named project's context.

- File, Move...

Moves the object(s) selected in the main view to another folder of the same type. E.g. move Resources between Projects, move Block Types between Libraries. This option is greyed out if there are no selected objects. A dialog box requests the destination of the move operation in terms that are suitable to the object being moved.

- File, Copy...

Copies the object(s) selected in the main view to another folder of the same type. E.g. copy Resources between Projects, copy Block Types between Libraries. A dialog box requests the destination of the copy operation in terms that are suitable to the object being moved.

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

**EI**

DOCUMENT NUMBER

NO NUMBER. For Information Only

SHT.

20

- File, Delete...  
Deletes the object(s) selected in the main view.
- File, Print...  
Prints the object(s) selected in the main view.
- File, Validate  
Validates the definition of the selected objects without performing a full build.
- File, Build...  
Builds the the selected objects into a (partially) runnable format.
- File, Close  
Closes the current main view. If the current view is other than the Project view the main view returns to the Project View. If the current view is the Project view the main view returns to an empty Project View (i.e. awaiting a File,New Project or File,Open Project).
- File, Exit...  
This command is the primary means of terminating the GCT session. A dialog box will request confirmation of this command, and further dialogs will prompt the user to save any unsaved edits (with options to save, discard edits or abort the exit procedure).

## Utilities Menu

This menu provides access to utilities which allow configuration of aspects of an application that are in addition to the control strategy, eg target-specific setup, etc. In the future this list will be expanded to incorporate more advanced features eg version control (see Future Utilities section below).

The Utilities pull-down menu contains the following items:

Library Browser
Windows...
Set Up

greyed out if there are no independent views

- Utilities,Library Browser  
This item invokes the Library Browse independent view
- Utilities,Windows...  
This item invokes a dialog box for controlling independent views. It is a convenience utility for managing screens with too many open windows. It includes a scroll list of all active independent windows and button for raising, lowering and killing the selected window(s).
- Utilities,Set Up  
This menu gives access to aspects of the GCT setup which will only occasionally need to be adjusted. For example, user preferences on default size or position of new windows.

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

21

## (view) Menu

This menu may be reset according to the current main view to offer view specific functions.

## Help Menu

This menu will give a number of access points to the help system. Initially a simple help system based on the current XVT help engine will be used but this will later be extended to a full hypertext help.

The Help pull-down menu contains the following items:

Index
Commands
Keyboard
Tutorial
About...

- Help, Index  
This command opens a window on the help system containing an alphabetical list of subjects addressed by the help system.
- Help, Commands  
This command opens a window containing a list of the entries in the top-level menu bar, from which the user can access help on any individual pull-down menu and any individual command which is available from the menu system.
- Help, Keyboard  
Whereas mouse users should find GCT almost totally intuitive, keyboard users will need special guidance on such subjects as navigating between windows, closing windows, and moving between controls in a dialog box. This command opens a window on the help system which gives keyboard users an alphabetical list of areas in which they may need assistance.
- Help, Tutorial  
This command will give first time users an interactive run-through of the facilities provided by GCT. The specification of this tutorial has yet to be defined.
- Help, About...  
This command displays a dialog box showing a copyright message and GCT version information.

### 7.2.3 Project View

Figure 7 illustrates the project view.

The project view provides a view of the Resources, Configurations, and Libraries (Project and Shared) associated with the project.

At any one time the Project View shows a scroll list of either the Resources (default), Configurations, or Libraries, together with "New" and "Open" buttons that operate on the current item type (i.e. create a new Resource or Open a selected Resource if the scroll list contains Resources).

From the Project View the user accesses items within a project and performs management functions on those items.

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER. For Information Only

SHT.

22

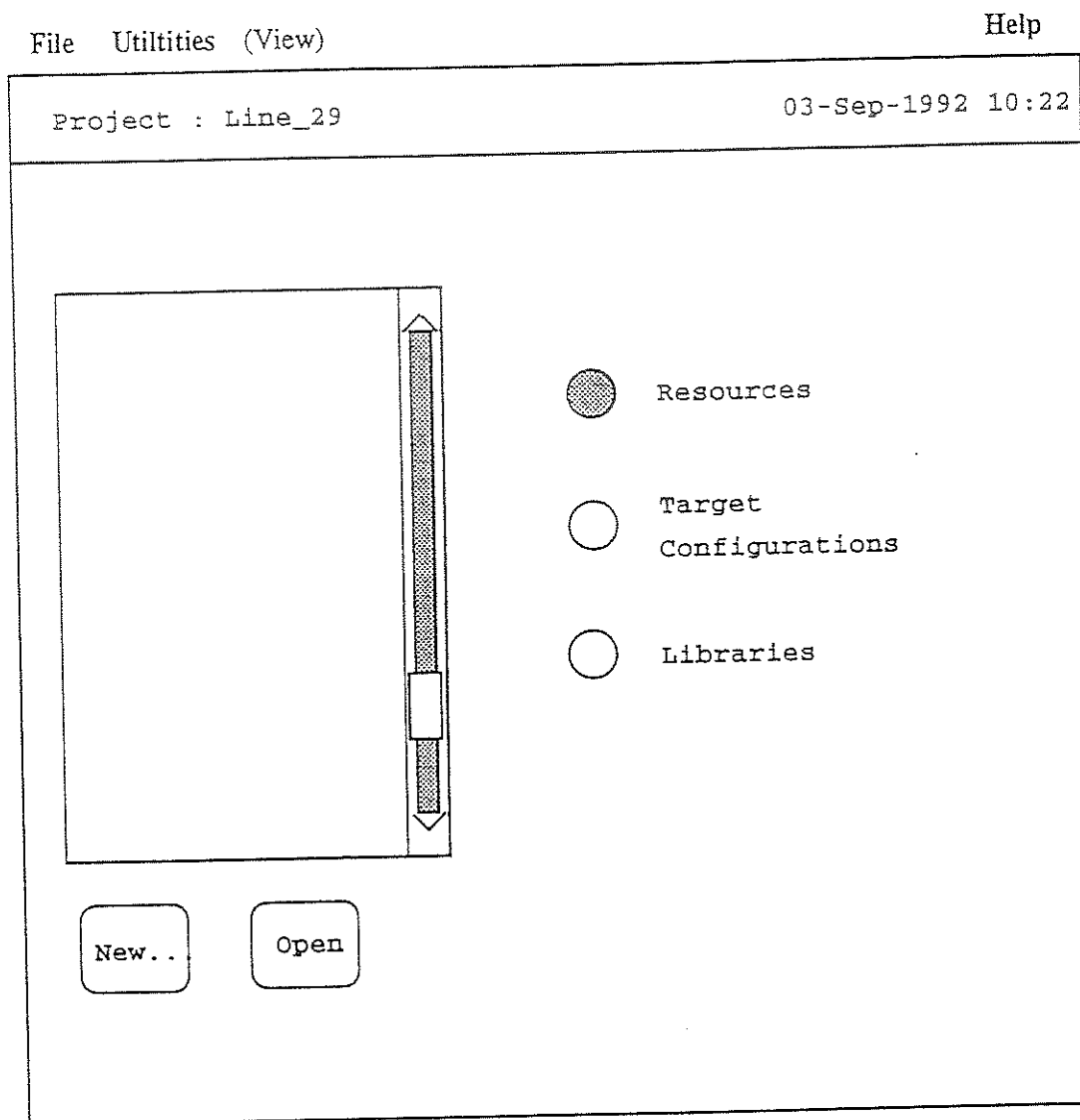


Figure 7: Main Project View

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Group Configuration Tool  Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER NO NUMBER, For Information Only	SHT. 23

### 7.2.4 Library View

The Library main view presents the contents of a single library. Option buttons allow the user to specify which block types will be included or inhibited from block list eg function block, service, function, etc.

From the library view blocks may be selected for library management functions. Also the edit may be launched of a new or existing block in an independent view.

### 7.2.5 Configuration View

The configuration view associates Resources with target nodes. Functionality includes

- Specifying which Resources are to run on which nodes.
- Defining network characteristics
- Downloading, starting, stopping Resources
- Launching a Resource editor in on-line mode.

### 7.2.6 FMC Target Support

A utility will be incorporated to allow configuration of target specific aspects of the FMC target system. This will be specified in more detail in conjunction with the FMC project.

## 7.3 Independent Views

Independent views exist in windows separate from, and largely independent of, the main view. There can be multiple active independent views, most of which will have their own menu bars.

### 7.3.1 Block View

The GCT block view will allow the display and modification of all application block types.

The block editor will consist of a one, two or three pane window and a menu bar, with each window area separated by a sash. The three panes will contain the interface, internals, and body views of a block.

The interface pane will provide a generic view of all block types, whilst the body view will be specific to the implementation of the block, eg SFC, ST, or in the future, FBD, spreadsheet or other editors as required.

In on-line mode, the block views will allow access to run-time values and the editing options will be greyed-out.

#### Block Menu Bar

The layout of the items in the menu bar will be as follows:

File	Edit	Options	Help
------	------	---------	------

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Group Configuration Tool	
		Provisional Functional Specification	
EUROTHERM © Copyright 1992 Eurotherm Limited	EI	DOCUMENT NUMBER NO NUMBER, For Information Only	SHT. 24

- File Menu

The File menu provides the means by which typical file-handling operations are performed on the block type. It contains the following items:

New...
Open...
Save
Save As...
Print...
Validate
Build...
Close

- File, New...

This launches the creation of a new block type in the current independent view. The user is prompted to save any changes to the existing block type in the view.

- File, Open...

This launches the Library Browser to select a new block type to edit in the current independent view. The user is prompted to save any changes to the existing block type in the view.

- File, Save

This item saves the current block to disk under its current name. Before saving to disk, a validation traverse will be performed (see File, Validate) to point out any errors to the user and, optionally, allow them to be corrected. It will be possible, however, to save an invalid block to disk.

It is possible that the present block will not yet have a name. In this case the Save command will perform a Save As (see below).

- File, Save As...

This command leads to a dialog box which enables the user to save the current block under a new name, and optionally launches the library browser to select the library in which the block is to be saved.

- File, Print...

Prints the block.

- File, Validate

Validates the definition of the block without performing a full build of the block.

- File, Build...

Builds the block into a format suitable for incorporation into a runnable Resource.

- File, Close

Closes the independent main view. The user is prompted to save any changes to the existing block type in the view.

- Edit Menu

The primary purpose of the Edit menu is to facilitate the cutting, copying and pasting of portions of editable components, and the undoing thereof.

The Edit pull-down menu contains the following items:

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> <small>© 1999 Eurotherm Limited</small>	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER. For Information Only	25



Undo
Cut
Copy
Paste
Delete
Search...

– Edit, Undo

Undo will not be supported initially. This command will undo the most recent Cut, Copy, Paste or Delete operation. It may be desirable for the text of the Undo item in the pull-down menu to be modified to show the type of operation which will be undone (e.g. Undo Paste). Consideration should also be given to the possibility of implementing a stack of edit operations to facilitate multiple undos.

– Edit, Cut

This command will remove the currently marked portions of an editable component and place them on the clipboard so that they may be subsequently pasted into another (or the same) editable component of the same type. This command will be greyed out whenever there is no marked area.

– Edit, Copy

This command will copy the currently marked portions of an editable component to the clipboard. This command will be greyed out whenever there is no marked area.

– Edit, Paste

This command will attempt to paste the clipboard contents into the current window.

In a text window, the pasted text will be inserted unless there is a marked area within the text window, in which case the pasted text will replace the marked portion.

In a graphical window the paste operation is effectively an overwrite, but if this involves deleting any portion of the existing window the user will be prompted to confirm this using a dialog box. At this point he will also have the option of aborting the entire Paste command.

– Edit, Delete

This command will remove the currently marked portions of an editable component without affecting the current clipboard contents.

– Edit, Search...

This command will not be supported initially. The command provides an entry point to an ST search and replace mechanism. A dialog box is displayed, having data entry fields for search text and replace text, and radio buttons for the scope of the search (e.g. current block, current program and entire configuration). Search text may be specified using regular expressions. There will be command buttons for Search, Replace and Cancel.

Once a search or replace operation has been initiated, editable components containing the search string will be displayed one after another. For the search operation a small dialog box will prompt for Next or Cancel, while for replace another small dialog will ask for confirmation with the options Yes, No, Global and Cancel.

• Options Menu

The Options menu will allow the user to configure options that are local in scope, such as grid settings for graphical editors.

• Help Menu

The Help menu provides a similar command set to that in the main menu's help pull-down, and will also provide access to the same help text as the main menu help. However, the initial entry points into the help system will be relevant to the local context.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	26

## Interface

The interface pane provides a generic representation of all application block types.

The interface editor would be support:

- Display interface declaration graphically (initially textually).
- Interactive help on specifying declaration type.
- Interactive help on data types
- Interactive help on declaration (e.g. edges, arrays, retain, etc).

## Internals

The internals pane provides a browsing facility for all components that are references within the body of the block.

The internals of a block with an SFC body would include steps, transitions and actions.

The internals editor would support:

- Display internals declaration graphically (initially textually).
- Interactive help on specifying declaration type.
- Interactive help on data types
- Interactive help on declaration (e.g. constant, arrays, retain, etc).

## Body

The representation of the body of a block will depend on how it was originally defined. When the body of a new block is first opened for edit, an option is given to select the body type. The selected editor will then be invoked to allow the body to be defined. Thereafter the body will retain its body type attribute and whenever it is selected for display or modification, the respective editor is again invoked.

In some cases it will be possible to display or transform a block body into a simpler type (usually to ST text), eg SFC to ST. It will not be possible to transform to a more complicated body type although default generation of graphics may be supported in future releases.

Initially only ST text and SFC body editors will be supported. The details of the interaction of these body views are described in later sections.

## Block Types

The following sub-sections describe features of the specific block types supported. In each case the editors that are appropriate for the definition of a body for a block of the given type are listed (those in brackets refer to future releases of GCT).

- Configuration:

The first release of GCT will not include the configuration level, in other words, configurations will consist of a single resource.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	27

- Interface: None.
- Internals: Resource declarations.
- Body may be defined in: ST, (FBD).
- Resource:
  - Interface: None.
  - Internals: Task, Service, Program, Function Block declarations.
  - Body may be defined in: ST, (FBD).
- Task:
  - Interface: None.
  - Internals: None.
  - Body may be defined in: Display list of task associations.
- Service:
  - Interface: Inputs, Outputs.
  - Internals: declarations (includes steps, transitions and actions if SFC body).
  - Body may be defined in: ST, SFC, (FBD).
- Program:
  - Interface: None.
  - Internals: declarations (includes steps, transitions and actions if SFC body).
  - Body may be defined in: ST, SFC, (FBD).
- Function Block:
  - Interface: Inputs, Outputs, Input\_Outputs.
  - Internals: declarations (includes steps, transitions and actions if SFC body).
  - Body may be defined in: ST, SFC, (FBD).
- Function:
  - Interface: Inputs, single output(implicit).
  - Internals: declarations.
  - Body may be defined in: ST, (FBD).
- Step:
  - Interface: Executing, Time, (Finished) parameters.
  - Internals: Action associations.
  - Body may be defined in: None.
- Transition:
  - Interface: None.
  - Internals: None.
  - Body may be defined in: ST, (FBD).
- Action:
  - Interface: Output.
  - Internals: None - uses declarations of POU in which action is defined. Will include steps and transitions if SFC body.
  - Body may be defined in: ST, SFC, (FBD).

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER

NO NUMBER, For Information Only

SHT.

28

### 7.3.2 Library Browser

The library browser view allows the user to browse and select blocks in libraries. The effect of selecting a block will depend on the context in which the browser is invoked. This may include launching an edit of the block type (in an independent view), enabling library management functions in the main view, etc.

The Library Browser operates on a list of libraries that is associated with a project. It will support the following:

- List libraries.
- List blocks within libraries.
- Allow the user to specify which block types will be included or inhibited from block list eg function block, service, function, etc.
- An existing or new block may be opened in a new block edit independent view.

### 7.3.3 Trace

A trace facility will provide a non-intrusive means of run-time diagnostics. A trace specification may be defined and downloaded and trace data can be collected from the target system and displayed. Trace specifications can be stored on file for later retrieval. Initially the trace utility will be based on the FMC trace specification but it is hoped that this will be of general use in the future.

## 8 SFC Editor

This editor, based on the IEC DIS 1131-3 Sequential Function Chart (SFC) graphical language (see [1]), will handle the configuration of steps, transitions, and alternate and parallel sequences.

Features of the GCT SFC editor will include:

- The SFC editor will present a palette of tool options.
- Placing of steps and transitions will be constrained to the SFC grid.
- Connecting lines will be constrained to orthogonal polylines. Bends in connections will be constrained to the SFC grid.
- A simple move facility will allow a single selected step, transition or connector to be moved. Any connections will be stretched.
- An object in the work area, eg a step or transition, may be opened for display and editing.
- Cut and Paste will be accessible via the menu bar. Initially this may be restricted to single SFC objects.

Tools available in the tool palette will include:

- Step

The Step tool will allow the placing of steps in the SFC work area.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	29

- Start Step

The start step tool will allow the start attribute of a step to be modified. A single body may have multiple SFC networks, but each network may only have a single start step. This tool may be used to support other attributes in the future.

- Transition

The Transition tool will allow the placing of transitions in the SFC work area.

- Connect

The connect tool will allow the wiring of steps and transitions.

A step may not be directly connected to another step. A transition may not be directly connected to another transition.

Option will be given for routing at connect time. Direct connection will result in auto-routing.

Define route from source to destination of connector.

Source and destination connection points will be highlighted when appropriate.

A connector will be allowed to hang loose a) when created, b) when object connected to is deleted.

If connections are made such that a transition has more than one following steps, the connections will become an activation.

If connections are made such that a transition has more than one preceding steps, the connections will become an rendez-vous.

If connections are made such that a step has more than one following transitions, the connections will become an alternate selection.

Connectors that represent Activation, Rendez-Vous, Alternate Divergence or Alternate Convergence constructs are known as complex connectors.

Fragments of a complex connector may be selectively deleted.

Move object results in connections being stretched (default routing).

More complex facilities for editing connectors may be added later: e.g. add points, move points, delete points. Initially editing will be done via create and delete.

- Delete

When the delete tool is active, steps, transitions or connectors may be selected for deletion. Fragments of a complex connector (Activation, Rendez-Vous, Alternate), may be selectively deleted depending on the position they are selected.

- Draw

A draw tool will be used to place textual or graphical information, freely within the work area (not supported initially).

- Group LATER

- Ungroup LATER

- Zoom in LATER

- Zoom out LATER

- Undo LATER

The UNDO facility will require a record to be kept of operations performed. Ideally a two level UNDO will be supported.

- Redraw

The redraw tool will allow refreshing of the work area.

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Group Configuration Tool

Provisional Functional Specification

**EUROTHERM**

© Copyright 1992 Eurotherm Limited

**EI**

DOCUMENT NUMBER  
NO NUMBER, For Information Only

SHT.

30

## 9 ST Editor

The ST editor will be a simple text editor based on the XVT facilities. ST text will be saved and blocks will be validated by the CDL Object Manager.

In fact, the ST editor will support the Configuration Definition Language (CDL) (see [2]) superset of the IEC DIS 1131-3 Structured Text (ST) language (see [1]), which uses ST for the specification of all run-time aspects. CDL incorporates extensions to ST for the specification of attributes that are used for display purposes as well as other important extensions to the language.

This editor will allow the textual definition of almost all aspects of a resource from the top layer of a resource down to a parameter attribute. Depending on the nature of the block being edited, the editor may in future assist the user by performing syntax checking and by providing templates for the textual definition.

It will be possible to import free format ST text from a file, into a block definition.

## 10 Future Utilities

### 10.1 Hierarchical Browser

The browser utility will be accessible at all times.

The browser will be hierarchical and will allow access to all levels of application data.

The browser will enable the user to browse definitions of all application entities:

- Configurations
- Resources
- Tasks
- Services
- Programs
- SFCs
- Steps within SFC POU
- Actions within SFC POU
- Function Blocks
- Functions
- Variables within POU

Multiple instances of the browser may be active concurrently.

The browser may be initialised to a given context and/or scope.

Additional operations on a selected object may be accessed via the browser: e.g. Open, New, Build, Verify, Delete.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool  Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited		<b>EI</b>	DOCUMENT NUMBER NO NUMBER, For Information Only  SHT. 31

## 10.2 User Screen Definition

This utility will allow the definition of textual user screens for debug purposes similar to those currently provided by the PC3000 Programming Software.

## 10.3 Diagnostics

Diagnostics utilities will show information eg on task overruns and system errors etc. Diagnostic utilities will probably be target specific.

## 10.4 Simulation and Debug

The ability will be provided to simulate the running of applications off-line, enabling the user to specify simulated behaviour of the physical I/O (which need not be present).

It will also be possible to access specialised debugging facilities when on-line: e.g. one-shot execution of tasks, SFCs, function blocks etc.

## 10.5 Source and Version Control

This facility will enable the user to save and manage different versions of his application source, tagging them with release numbers when required, and will allow him to revert to any of the previous versions at any time.

## 10.6 Additional Target Support

Additional utilities will provide support specific to a given target system (e.g. Production Orchestrator, PC3000, T1000, etc).

In the case of PC3000, for example, this facility will enable the user to specify the required configuration of hardware modules within any of the PC3000 resources in the system. This facility will act in a more 'intelligent' manner than the equivalent provision of the current PS: the user will be allowed to allocate 'virtual' I/O when laying out the configuration, the hardware management utility later showing him the virtual I/O which has been assigned and allowing him to match this to actual I/O modules as required.

Whatever the target system, this facility will include only those features which are not supported by the standard GCT utilities.

## 10.7 FBD Editor

The IEC DIS 1131-3 Function Block Diagram (FBD) graphical language (see [1]) will be used to configure components at different levels within the configuration hierarchy (e.g. configuration, resource, program, function block, function). In practice there will be a single FBD editor that will operate in different modes according to context.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool	
		Provisional Functional Specification	
<b>EUROTHERM</b> © Copyright 1992 Eurotherm Limited	<b>EI</b>	DOCUMENT NUMBER	SHT.
		NO NUMBER, For Information Only	32

## 10.8 Other Editors

Other editors may be incorporated into the GCT framework eg the MICS spreadsheet editor.

## 11 GCT Topics

The following sub-sections summarise the manner in which GCT supports a number of topic areas through its various views.

### 11.1 Help

Interactive Help on all aspects of GCT interaction will be built in to all GCT views.

### 11.2 Cut And Paste

GCT cut and paste will be based on the XVT cut and paste facilities. XVT employs a single clip board which is capable of carrying multiple representations of objects. In the future it will be possible to cut and paste between different editor types, eg an ambitious example would be from a spreadsheet editor to an SFC editor. Initially simple cut and paste of text will be supported.

### 11.3 Application Documentation

GCT will support the ability to add descriptive (initially textual, but later to include graphical) information to application components.

A hardcopy of application documentation may be generated for each Configuration, Resource or Block definition. This will be accessible via the Print command in the Project pull-down of the main menu or the Block pull-down of individual view menus.

### 11.4 Run-Time Facilities

GCT will enable the user to manage the building, downloading and executing of applications. This will be accessible via the menu bar options of the Project and Library (build only) views.



CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Group Configuration Tool  Provisional Functional Specification	
EUROTHERM  © Copyright 1992 Eurotherm Limited	EI	DOCUMENT NUMBER  NO NUMBER, For Information Only	SHT.  33



File Utilities (View)

Help

Project : Line\_29

03-Sep-1992 10:22

--- Resource List ---

Diameter_Cntrl
Extruder_Cntrl
Extruder_Maint
<b>Hopper</b>
Winder

- ☒ Resources
- ☐ Target Configurations
- ☐ Libraries
- ☐ Graphics

New...

Open

File Edit Options Help

SFC Editor Block : Thermal\_Detect

Tools

Inl Error  
PV  
SetPoint  
Cntrl\_OP

ACTIONS  
Detect  
StratUp

File Edit Options Help

Panel Graphics Editor File : Feeder\_View

File Edit Options Help

Spreadsheet Editor Block : Thermal\_Cntrl

File Utilities (View)

Help

New Project...

Open Project...

Move ...

Copy ...

Delete

Print ...

Validate

Build ...

Close

Exit

List ---

Winder

New...

Open

03-Sep-1992 10:22



Resources



Target  
Configurations



Libraries



Graphics