

Cliente ENEL Produzione**Oggetto** Guida all'uso dei sistemi per la modellizzazione dei sistemi di automazione in ambiente ALTERLEGO**Ordine** ICI 20/I/2006 - Contratto tra Enel.NewHydro e Enel Produzione del 28 Ottobre 2004**Note** Rev. 0

La parziale riproduzione di questo documento è permessa solo con l'autorizzazione scritta del CESI.

N. pagine 53 **N. pagine fuori testo****Data** 03/04/2006**Elaborato** SEL – SC Francesco Rizzo**Verificato** SEL – SC Massimo Pozzi**Approvato** SEL Antonio Ardito

Indice

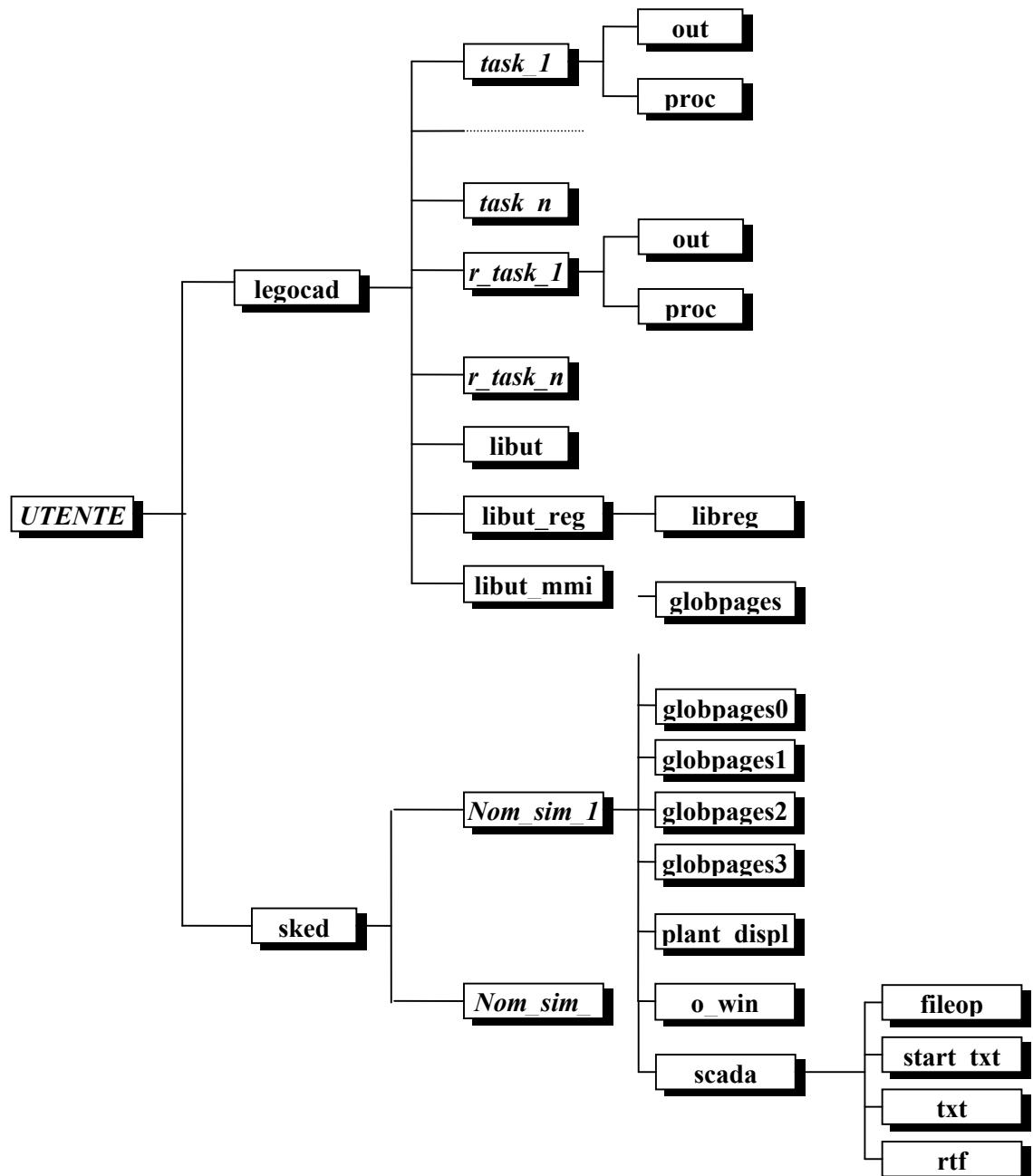
1 SCOPO DEL DOCUMENTO	4
2 STRUTTURA DELLE DIRECTORY	4
3 REGO TASK	5
4 CREAZIONE DI UNA PAGINA REGO	6
4.1 Istanziazione e configurazione di moduli REGO	7
4.2 moduli funzionali	8
4.2.1 ICI CONTROL DRIVE	8
4.2.2 CC - CONTROL INTERFACE	9
4.2.3 ICI - MOTOR/SOLENOID VALVE	10
4.2.4 ICI - ACTUATOR	11
4.2.5 MASTER CONTROLLER	12
4.2.6 MEASURE	13
4.2.7 SETP ADJUSTER	14
4.2.8 ACTUATOR SWITCH-OVER	15
4.2.9 SUB GROUP CTRL	16
4.2.10 GROUP CTRL	17
4.3 Connessione tra variabili LEGO	18
4.3.1 Connessione di variabili in una pagina REGO	18
4.3.2 Connessione di variabili tra pagine REGO della stessa TASK	18
4.3.3 Connessione di variabili tra pagine REGO di TASK diverse	18
4.3.4 Connessione tra variabili di regolazione e di processo	18
4.3.5 Connessione di allarmi del tipo A, P, T	18
5 INTEGRAZIONE DI TASK DEL SISTEMA ELETTRICO	19
5.1 Motori con modello fisico (MASI+GIPO)	Errore. Il segnalibro non è definito.
5.2 Motori con modello semplificato	Errore. Il segnalibro non è definito.
6 SISTEMA MMI	30
6.1 Pagine MMI	30
6.2 Stazioni di controllo/supervisione	30
6.3 Sistema SCADA	32
6.4 Trend variabili	33
7 CREAZIONE DI UN SIMULATORE	34
8 DOCUMENTAZIONE COMPLEMENTARE	38
9 DATA-BASE DELL'AUTOMAZIONE	40
9.1 ARCHITETURA	40
9.2 Descrizione dell'interfaccia utente	43
9.2.1 Tabelle di riferimento	44
9.2.2 Tabelle dati	45
9.2.3 Tabelle Control Window	48
9.3 Aggiornamento della documentazione sullo stato del simulatore	49
9.4 Aggiornamento automatico delle tabelle Siemens pages e Rego pages	49
9.5 Gerarchia di plant displays e control windows	50
9.5.1 Concetto di gerarchia	50

9.5.2	Assegnazione della gerarchia	51
9.6	Documentazione del progetto.....	52
9.6.1	Documentazione delle Control Windows.....	52
9.6.2	Documentazione degli allarmi.....	52
9.6.3	Specifiche funzionali sui diagrammi funzionali.....	53
9.6.4	Specifiche funzionali sul MMI.....	53
9.6.5	Test report.....	53

1 SCOOPO DEL DOCUMENTO

L'obiettivo del documento è quello di dare le principali indicazioni utili alla creazione/modifica delle task di regolazione e di Man Machine Interface (MMI) in ambiente ALTERLEGO. Particolare risalto sarà dato all'utilizzo della libreria TELEPERM, emulazione dell'omonimo sistema di controllo SIEMENS.

2 STRUTTURA DELLE DIRECTORY



3 REGO TASK

Contengono la struttura del sistema di controllo che viene realizzata mediante il configuratore grafico “config” messo a disposizione dall’ambiente ALTERLEGO.

In ogni task devono essere inizialmente presenti i seguenti file:

- Context.ctx contiene tutte le informazioni sia sulle librerie utilizzate sia sulle pagine di controllo create
- FileColorDefault definisce i colori di background direttamente disponibili
- FileFontsDefault definisce i font direttamente disponibili

Questi files vengono creati automaticamente la prima volta che viene lanciato l’applicativo privi di qualsiasi dato di configurazione che deve quindi essere immessa manualmente. Si consiglia quindi di copiare i files FileColorDefault e FileFontsDefault da una task già esistente. È infine parimenti opportuno importare una pagina di regolazione in modo da mantenere il lay-out base delle pagine TELEPERM.

```
*animatedIconLibraries: /usr/users/cfb_int/legocad/libut_reg/libreg
*page_list: \ rego C001 C002 C003 C004 C005 C006 C007 C008 C009 C010 C011 C012 C013 C014 C015 C016 C017
*iconlib_label: \ LIB_FX LIB_GC LIB_IO LIB_LOG LIB_MAT LIB_MEASURE LIB_EQ LIB_SGC LIB_SLC std
*hostNameS:
*iconlib_num: 10
*displayList:
*hostName:
*pag_num: 51
*description: Condensato
*pages: ./
*simulator: ./
*numDisplay: 0
*objectLibraries: /usr/users/cfb_int/legocad/libut_reg/libreg
*nextTagPag: 1H
*iconlib_list: \ LIB_FX LIB_GC LIB_IO LIB_LOG LIB_MAT LIB_MEASURE LIB_EQ LIB_SGC LIB_SLC std
```

```
asedit - FileColorDefault.ok
File Edit Search Tools
[*drawnButton7.background: #94949393a1a1
*drawnButton8.background: #000000000000
*drawnButton5.background: #787866661515
*drawnButton3.background: #chchcbcbcb
*drawnButton6.background: #2828aaaaffff
*drawnButton10.background: #e1e1a2a24141
*drawnButton2.background: #fffffebe0000
*drawnButton4.background: #ffffffff0000
*drawnButton12.background: #00000000c5c5
*drawnButton11.background: #ffffffff9696
*drawnButton1.background: #ffffd2d20000
*drawnButton9.background: #0000d2d29292
```

```
asedit - FileFontsDefault.ok
File Edit Search Tools Help
*FontSample2.fontList: -adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1
*FontSample2.nomeFont: -adobe-helvetica-bold-r-normal--14-140-75-75-p-82-iso8859-1
*FontSample3.fontList: -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-1
*FontSample3.nomeFont: -adobe-helvetica-bold-r-normal--18-180-75-75-p-103-iso8859-1
*FontSample4.fontList: -adobe-helvetica-bold-r-normal--17-120-100-100-p-92-iso8859-1
*FontSample4.nomeFont: -adobe-helvetica-bold-r-normal--17-120-100-100-p-92-iso8859-1
*FontSample1.fontList: -adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1
*FontSample1.nomeFont: -adobe-helvetica-bold-r-normal--12-120-75-75-p-70-iso8859-1
*FontSample5.fontList: fixed
*FontSample5.nomeFont: fixed
```

4 CREAZIONE DI UNA PAGINA REGO

Con il comando **config** si attiva il programma omonimo di configurazione delle pagine REGO.

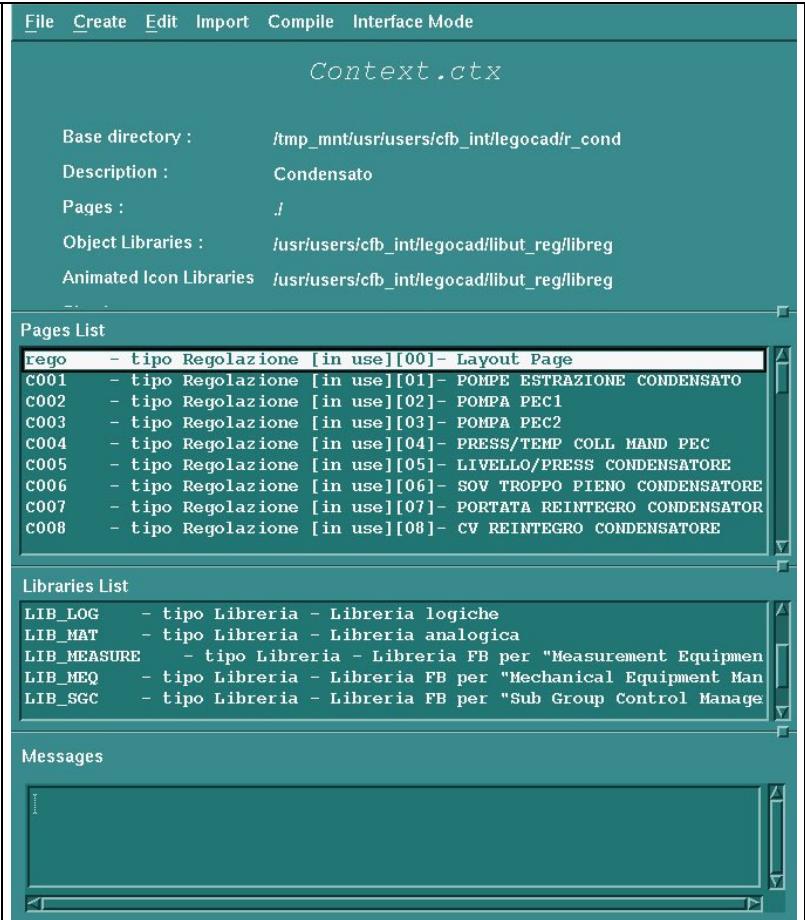
A seconda della configurazione definita precedentemente nel file Context.ctx verranno visualizzate nella finestra “Libraries List” l’elenco delle librerie successivamente disponibili per le operazioni di costruzione delle strutture di regolazione. Le principali funzioni sono:

menù “Create” per la creazione di una area di lavoro; di norma si preferisce definire un layout standard di pagina REGO a seconda del progetto in corso e quindi duplicare questa pagina tramite l’apposito comando.

Menù “Import” permette di importare pagine REGO o librerie presenti in altri progetti.

Menù “Compile” per le operazioni di compilazione e creazione di pagine grafiche.

Menù **Interface** per le operazioni di connessione variabili tra pagine diverse.



4.1 Istanziazione e configurazione di moduli REGO

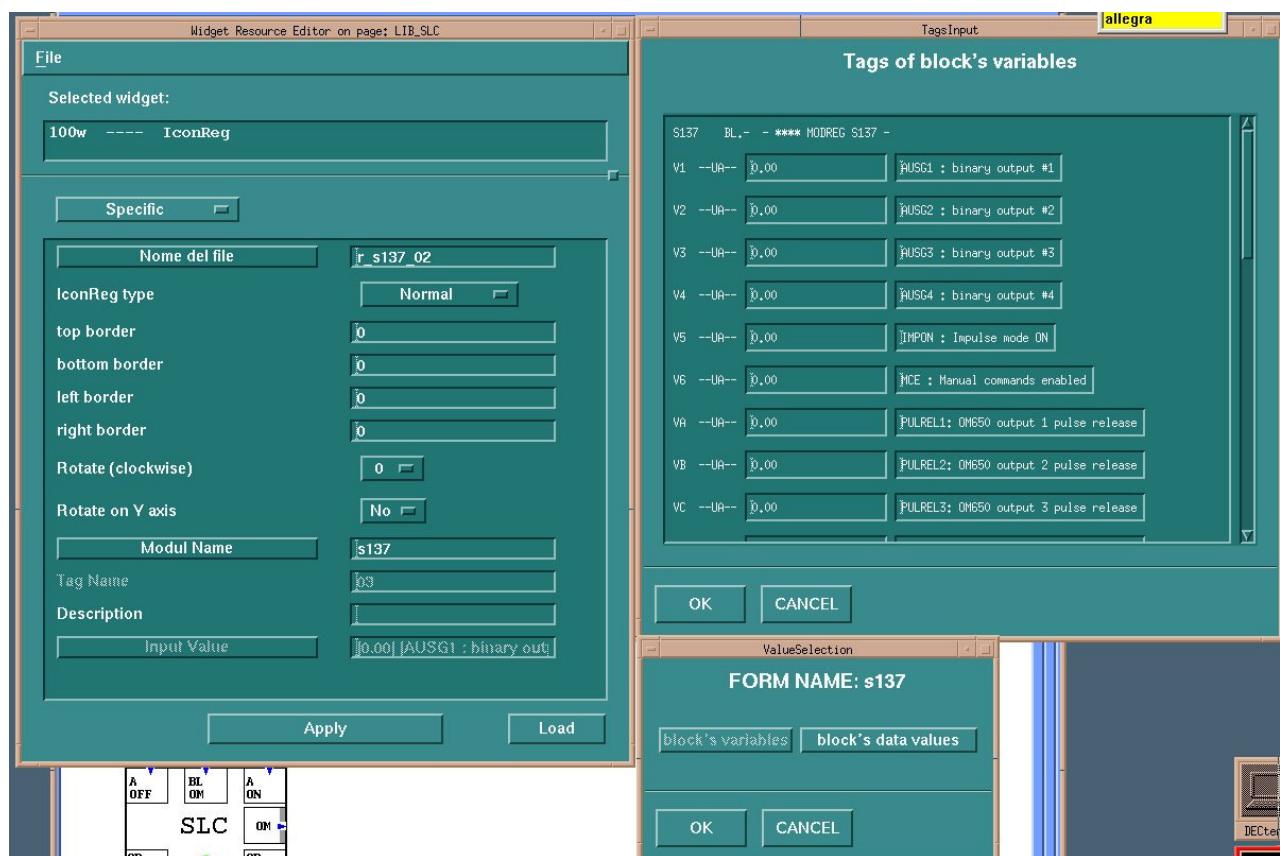
Una volta aperta la pagina REGO precedentemente creata si passa alla istanziazione dei moduli di regolazione.

Una volta individuata ed aperta la libreria che contiene il modulo tramite il Menù “Add Icon” è sufficiente trascinare con il tasto centrale del mouse il modulo scelto all’interno dell’area di lavoro.

Una volta istanziato il modulo si procede alla configurazione del modulo, attraverso il **resource editor** che si attiva cliccando il tasto destro sul modulo selezionato.

Le variabili da configurare dipendono dal modulo che si sta utilizzando e si dividono, principalmente, in due categorie:

- variabili di parametrizzazione, riguardano i valori che caratterizzano il funzionamento di un modulo, come ad esempio:
 - ❖ costante proporzionale o integrale di un regolatore
 - ❖ tempo di corsa di un attuatore
 - ❖ soglie di allarme di una misura
- variabili di inizializzazione, riguardano le variabili che determinano lo stato di partenza del modello. La corretta attribuzione di tali valori è molto importante (specialmente nel caso di modelli di una certa complessità) per il buon funzionamento di un simulatore. Si tratta ad esempio di:
 - ❖ richiesta di posizione di una valvola
 - ❖ variabili di processo controllate
 - ❖ funzioni di controllo con memorizzazione di uno stato



4.2 moduli funzionali

Vengono di seguito descritti i principali moduli funzionali presenti nella libreria TELEPERM.

4.2.1 ICI CONTROL DRIVE

È la configurazione tipica per l'attuazione di valvole con comando in velocità. I componenti utilizzati sono:

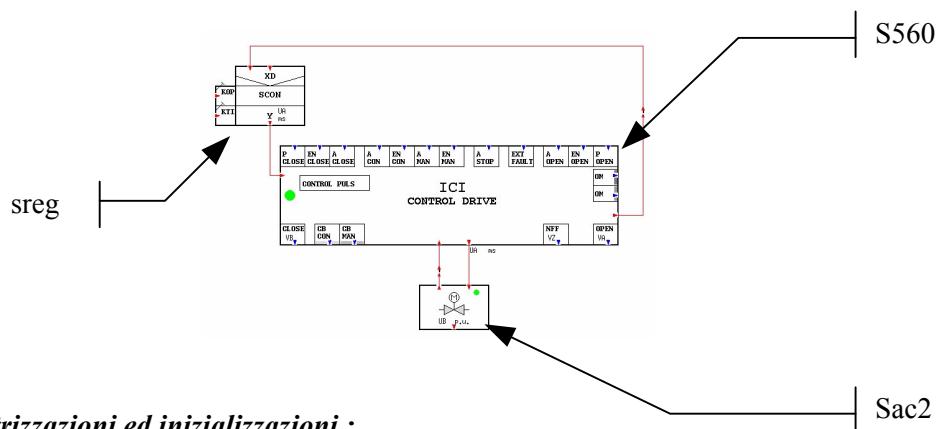
Step controller (sreg) : è il regolatore; genera, se in assetto automatico, un segnale di uscita v elaborazione

$$\text{dell'errore } \varepsilon \text{ della variabile controllata} : v = K_p * (\varepsilon + \frac{1}{T_i} \int \varepsilon * dt)$$

Control Drive (s560) : è il cassetto di potenza; genera il comando verso l'attuatore seguendo, in assenza di comandi di protezione (quindi a più alto livello gerarchico), le richieste del regolatore

Actuator (sac2) : è l'attuatore, cioè l'organo che meccanicamente muove lo stelo della valvola o dell'accoppiamento del componente a cui è collegato

Rego Page

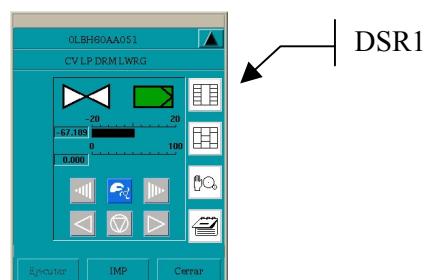


Principali parametrizzazioni ed inizializzazioni :

sreg	J0	PAUS	"Disable P component"
	J1	IAUS	"Disable I component"
	IA	KOP	"Gain factor (P component)"
	IB	KTI	"Time constant (I component)"
s560	VV	@#K@KKS_XF51	"Fault Switch to Man"
	VW	@#K@KKS_XC86	"Protec/fallo Pos Fin"
sac2	UB	@#K@KKS	Position setp (p.u.)
	IA	TRAVTIME	Travel time
	IV	@#K@BUSBAR	Voltage

Stazione di Comando

DSR1 Valve
DSR3 Damper



DSR1

4.2.2 CC - CONTROL INTERFACE

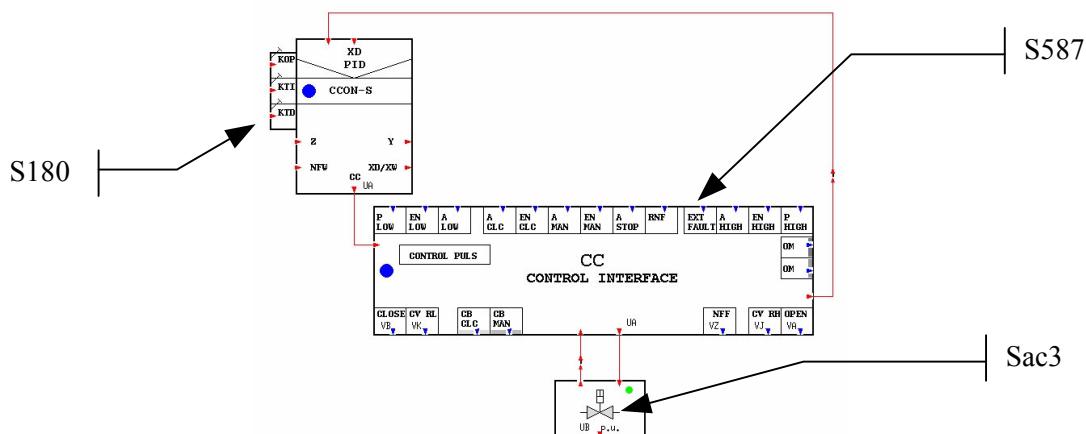
Rego Page

È la configurazione tipica per l'attuazione di valvole con comando in posizione. I componenti utilizzati sono:

Continuos controller (s180) : è il regolatore; genera, se in assetto automatico, un segnale di uscita θ
elaborazione dell'errore ϵ della variabile controllata : $\theta = K_p * (\epsilon + \frac{1}{Ti} \int \epsilon * dt + \frac{Kd}{Ktd} \frac{\partial \epsilon}{\partial t})$

Control Interface (s587): è il cassetto di potenza; genera il comando verso l'attuatore seguendo, in assenza di comandi di protezione (quindi a più alto livello gerarchico), le richieste del regolatore

Actuator (sac3) : è l'attuatore, cioè l'organo che meccanicamente muove lo stelo della valvola o dell'accoppiamento del componente a cui è collegato

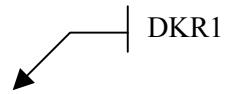
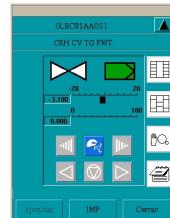


Principali parametrizzazioni ed inizializzazioni:

sreg	J0	PAUS	"Disable P component"	[1]
	J1	IAUS	"Disable I component"	[1]
	J2	DAUS	"Disable D component"	[0]
	IA	KOP	"Gain factor (P component)"	[]
	IB	KTI	"Time constant (I component)"	[]
	IC	KTD	"Derivative action time constant"	[]
	ID	T1	"Attenuation time constant"	[]
	IH	YO	"Upper limit for Y"	[..]
	II	YU	"Lower limit for Y"	[0]
	UA	Y	"Manipulated Variable"	[]
	VC	ISTREGELN	"Closed loop ctrl mode"	[1]
	VD	ISTHAND	"MAN/Open loop ctrl mode"	[0]
s560	VA	IZA	"Actual state OPEN/ON"	[..]
	VB	IZZ	"Actual state CLOSED/OFF"	[..]
	VV	@#K@KKS_XF51	"Fault Switch to Man"	"llega"
	VW	@#K@KKS_XC86	"Protec/fallo Pos Fin"	"llega"
	UA	Y	"Manipulated Variable"	[]
	I1	TYSH	"Rump Up time with protection"	[]
	I2	TYST	"Rump Down time with protection"	[]
	I3	TYH	"Rump Up time"	[]
	I4	TYT	"Rump Down time"	[]
	I5	OG	"Upper Limit Value"	[100]
	I6	UG	"Lower Limit Value"	[0]
sac2	UA	OUTPC	Lift position %	[..]
	UB	@#K@KKS	Lift position pu	[..]
	IA	TRAVTIME	Travel time	
	IV	@#K@BUSBAR	Voltage	

Stazione di Comando

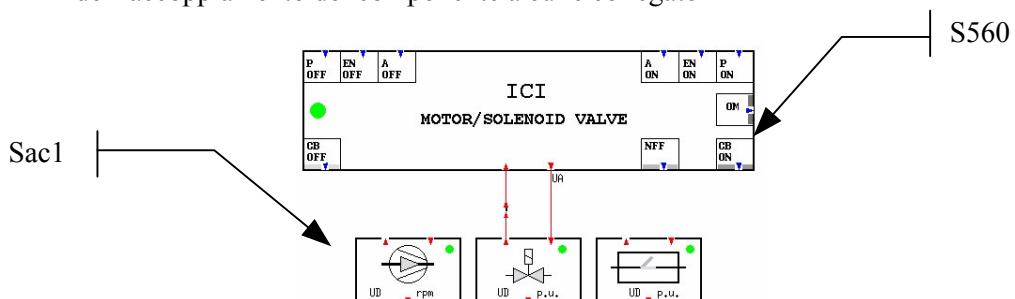
DKR1 Valve
 DKR3 Damper
 DKR7 Valve with lift position

**4.2.3 ICI - MOTOR/SOLENOID VALVE****Rego Page**

È la configurazione tipica per l'attuazione di valvole a solenoide. I componenti utilizzati sono:

Motor/Solenoid Valve (s560) : è il cassetto di potenza; genera il comando verso l'attuatore in base ai comandi logici di Apri/Chiudi ricevuti

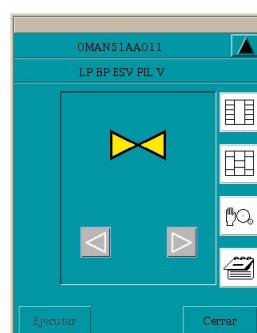
Actuator (sac1) : è l'attuatore, cioè l'organo che meccanicamente muove lo stelo della valvola o dell'accoppiamento del componente a cui è collegato

**Principali parametrizzazioni ed inizializzazioni:**

s560	VA	IZA	“Voith position %”	[..]
	VB	IZZ	“Actual state CLOSED/OFF”	[..]
	VW	@#K@KKS_XB86	“Protec/fallo Pos Fin”	“illega”
sac1	UA	YVOITHPC	Voith position % (per pompe)	[..]
	UB	YDRIVEPC	Motor drive speed % (pos for valve)	[..]
	UC	YPUMPPPC	Pump speed %	[..]
	UD	@#K@KKS	Pump scaled speed rad/s (pos for SV)	[..]
	ID	OFF_VAL	Motor drive speed in OFF status rad/s	[..]
	IE	ON_VAL	Motor drive speed in ON status rad/s	[..]
	IH	ON_TIME :	Motor drive start-up time	[..]
	II	OFF_TIME :	Motor drive shut-down time	[..]
	IV	@#K@BUSBAR	Voltage	[..]

Stazione di Comando

DEV1 Valve
 DEV3 Damper
 DEM2 Pump
 DEM4 Compressor
 DEM5 Electrical breaker
 DEM6 Motor



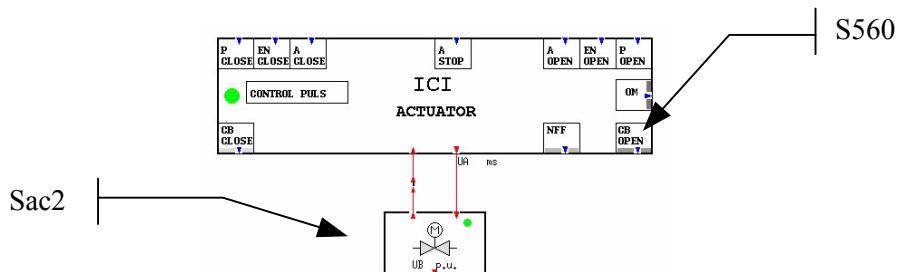
4.2.4 ICI - ACTUATOR

Rego Page

È la configurazione tipica per l'attuazione di valvole motorizzate. I componenti utilizzati sono:

Actuator (s560) : è il cassetto di potenza; genera il comando verso l'attuatore in base ai comandi logici di Apri/Chiudi ricevuti

Actuator (sac2) : è l'attuatore, cioè l'organo che meccanicamente muove lo stelo della valvola o dell'accoppiamento del componente a cui è collegato

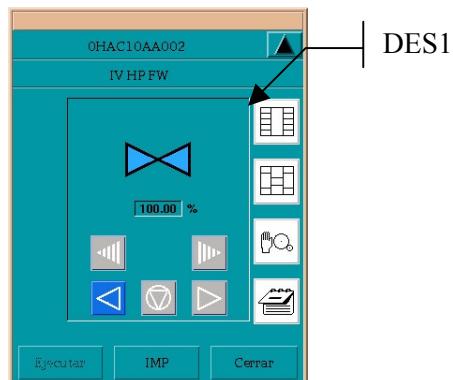


Principali parametrizzazioni ed inizializzazioni :

s560	VW	@#K@KKS_XB86	"Protec/fallo Pos Fin"	"llega"
sac2	UB	@#K@KKS	Position setp (p.u.)	
	IA	TRAVTIME	Travel time	
	IV	@#K@BUSBAR	Voltage	

Stazione di Comando

DES1 Valve
DES3 Damper



4.2.5 MASTER CONTROLLER

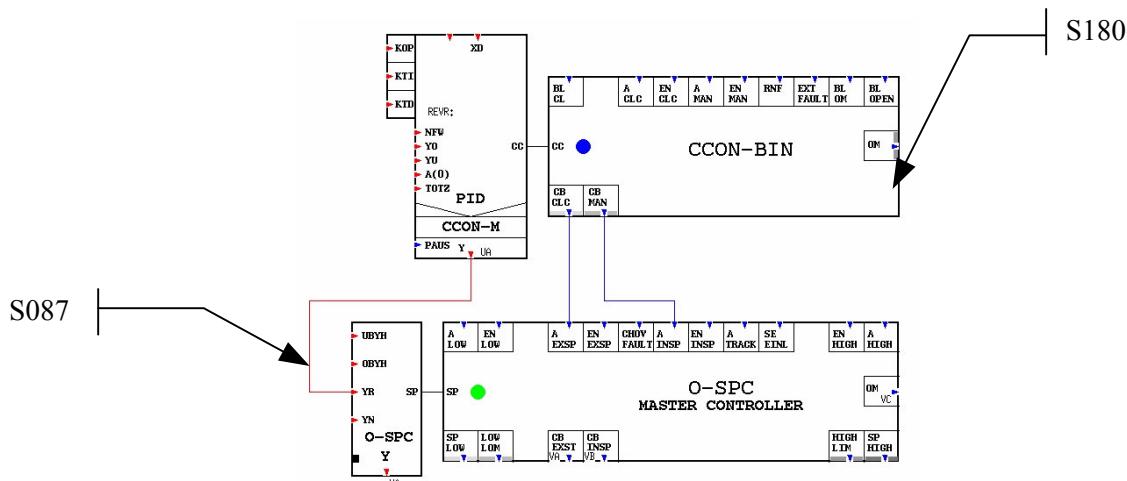
Rego Page

È la configurazione tipica per la realizzazione di funzioni di controllo non direttamente collegate ad un attuatore, tipicamente per le regolazioni a livello gerarchico più alto. I componenti utilizzati sono:

CCON-BIN (s180) : è il regolatore; genera, se in assetto automatico, un segnale di uscita θ elaborazione

$$\text{dell'errore } \varepsilon \text{ della variabile controllata} : \theta = K_p * (\varepsilon + \frac{1}{T_i} \int \varepsilon * dt + \frac{K_d}{Ktd} \frac{\partial \varepsilon}{\partial t})$$

O-SPC (s087) : è la memoria di uscita, necessaria nelle fasi di transizione AUT↔MAN o di TRACK



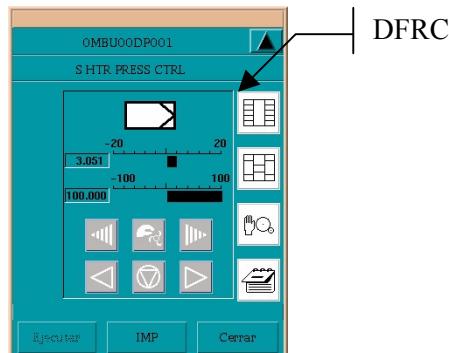
Principali parametrizzazioni ed inizializzazioni:

s180 (s176)	VI	@#K@KKS_XF52	“Fault Switch to Man”	“llega”
	J0	PAUS	“Disable P component”	
	J1	IAUS	“Disable I component”	
	IA	KOP	“Gain factor (P component)”	
	IB	KTI	“Time constant (I component)”	

s087 Nessun allarme

Stazione di Comando

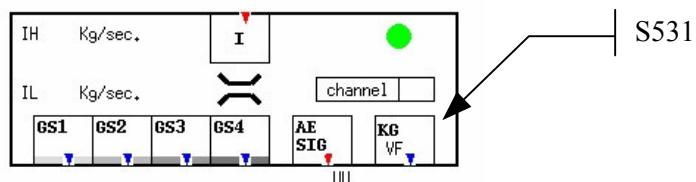
DFRS/DFRC



4.2.6 MEASURE

Rego Page

Meas (s531) : è il trattamento misure, genera un valore scalato tra i valori minimi e massimi del misuratore, nonché le varie soglie ed il segnale di anomalia misura

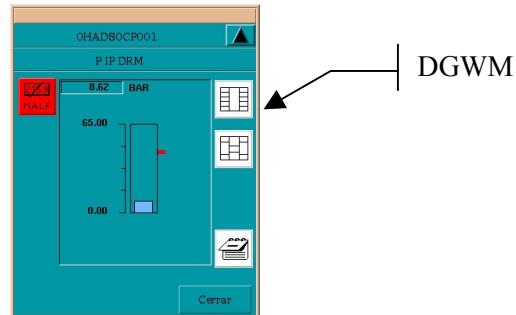


Principali parametrizzazioni ed inizializzazioni:

s531	VF	@#K@KKS_XZ01	"Falla en canal"	"llega"
	IL	IL : x scale low limit		
	IH	IH : x scale high limit		
	I1	GW1 : LIMIT VALUE #1		
	I2	GW2 : LIMIT VALUE #2		
	I3	GW3 : LIMIT VALUE #3		
	I4	GW4 : LIMIT VALUE #4		
	J1	GSB1 : TRIGGERING LIMIT DIRECTION #1		
	J2	GSB2 : TRIGGERING LIMIT DIRECTION #2		
	J3	GSB3 : TRIGGERING LIMIT DIRECTION #3		
	J4	GSB4 : TRIGGERING LIMIT DIRECTION #4		

Stazione di Comando

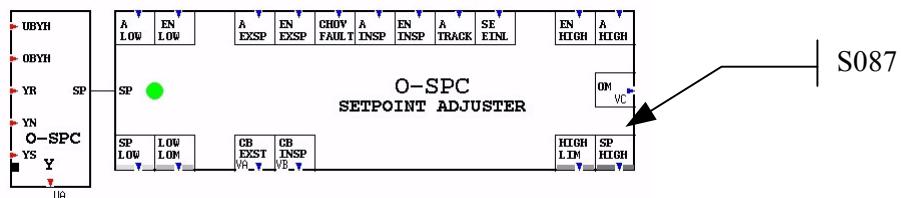
DGWM/DGMW_P



4.2.7 SETP ADJUSTER

Rego Page

O-SPC (s087) : è il Set-Point generator, cioè il blocco che genera una variabile di riferimento, secondo la richiesta dell'operatore o delle logiche previste



Principali parametrizzazioni ed inizializzazioni:

s087

VC

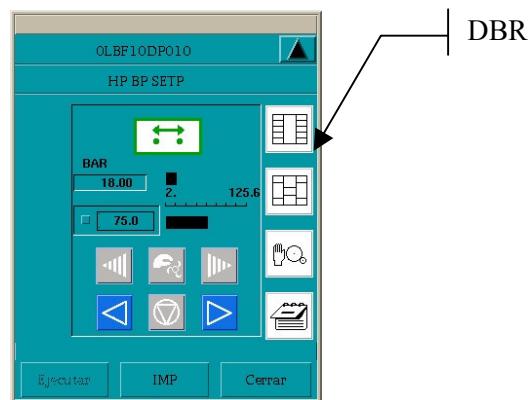
@#K@KKS_XZ01

“Fault Switch to IS”

“llega”

Stazione di Comando

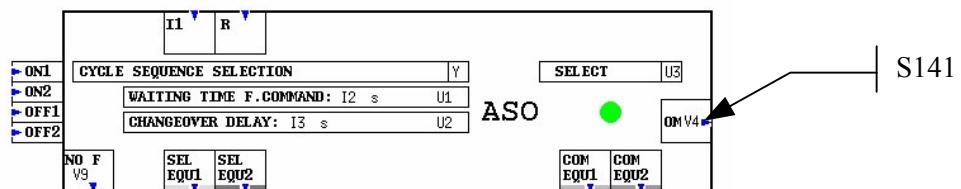
DBR



4.2.8 ACTUATOR SWITCH-OVER

Rego Page

ASO (s141) : è un sistema complesso che contiene una serie di logiche (partenza automatica, switch-over, ecc.) che presiedono alla gestione di due (o più) attuatori funzionanti in parallelo

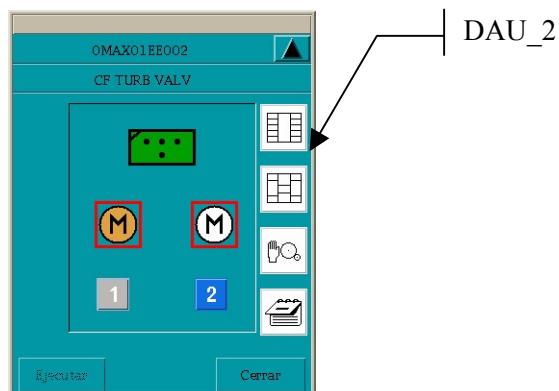


Principali parametrizzazioni ed inizializzazioni:

s141 V9 @#K@KKS_XA05 “Fallo” “llega”

Stazione di Comando

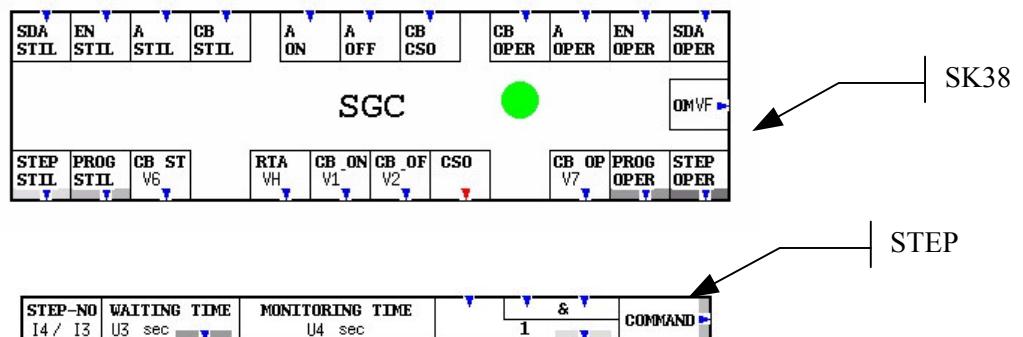
DAU_2/DAU_3



4.2.9 SUB GROUP CTRL

Rego Page

SGC (sk38)/step : è un sistema che gestisce sequenze complesse di operazioni



Principali parametrizzazioni :

s138
step

VF

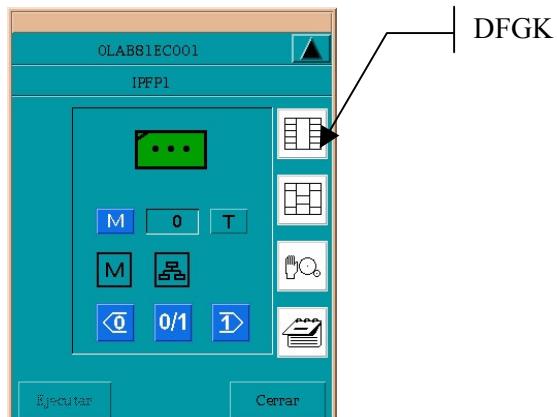
@#K@KKS_XZ01

“Laufzeit Schritt n”

“llega?”

Stazione di Comando

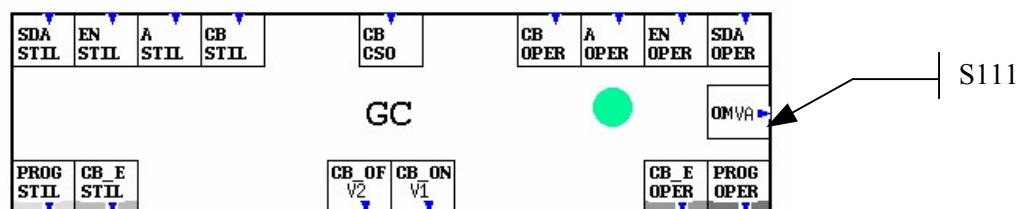
DFGK



4.2.10 GROUP CTRL

Rego Page

GC (s111) : è un sistema che gestisce le logiche di partenza e switch-over di due (o più) sequenze funzionanti in parallelo



Principali parametrizzazioni ed inizializzazioni:

s111

VF

@#K@KKS_XZ01

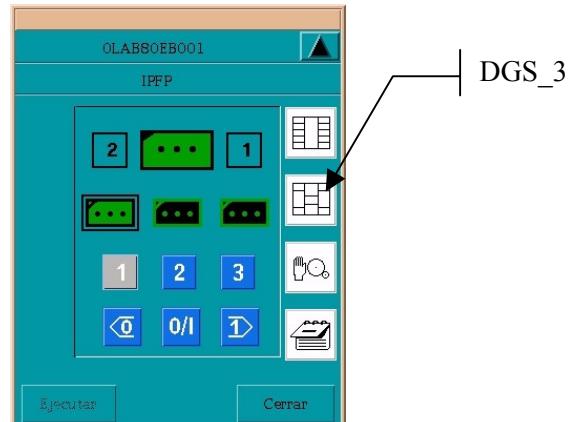
“Anf. n. erf. BETRIEB”

“llega”

Stazione di Comando

DGS_2

DGS_3



4.3 Connessione tra variabili LEGO

4.3.1 Connessione di variabili in una pagina REGO

La connessione tra moduli o icone di I/O si effettua graficamente unendo le ingresso ed uscita da collegare. Per farlo selezionare, dal Menù **Options** della pagina REGO la voce **Connected mode**.

4.3.2 Connessione di variabili tra pagine REGO della stessa TASK

La connessione tra icone di I/O di pagine REGO facenti parte della medesima task si effettuano come di seguito descritto:

- Salvare e chiudere tutte le pagine REGO aperte
- Selezionare l'opzione **Interface Mode** nella finestra **config**
- Aprire le pagine in cui devono essere eseguiti i collegamenti (se ne possono aprire fino ad un massimo di cinque contemporaneamente)
- Cliccare sulle frecce delle due icone di I/O
- Salvare il **config** (chiudendo la modalità Interface Mode le connessioni eseguite vengono salvate nel file Connessioni.reg)

Nota: qualora si debba eliminare un modulo a cui è collegata direttamente una icona interfaccia oppure il collegamento stesso, è necessario, prima di cancellare tale modulo, eliminare il collegamento con altre pagine.

4.3.3 Connessione di variabili tra pagine REGO di TASK diverse

La connessione di variabili tra pagine REGO si effettua mediante le icone I/O gialle, configurando nel campo descrizione una stringa del tipo @#K@KKSpartenza#Nomevariabile#KKSarrivo# che verrà in seguito interpretata dal programma “connex”

Questa stringa andrà configurata mediante il **resource** editor sia nella variabile di uscita che in quella di ingresso nel modulo “SAIO/SLIO” che si devono collegare.

4.3.4 Connessione tra variabili di regolazione e di processo

La connessione di variabili con TASK LEGO si realizza inserendo una stringa del tipo @#K@KKS tramite il **resource** editor nel campo descrizione della variabile che deve essere collegata.

4.3.5 Connessione di allarmi del tipo A, P, T

L'inserimento di allarmi in una pagina REGO, che si devono connettere al sistema **scada**, si effettua come descritto nel par. 4.3.3, utilizzando però l'icona di I/O azzurra e configurando il campo descrizione con una stringa del tipo @#K@KKS_nomesegnale

Nota: gli allarmi di tipo F si configurano nel blocco funzionale come indicato nel par. 4.3.4

5 INTEGRAZIONE DI TASK DEL SISTEMA ELETTRICO

L'ambiente di simulazione ALTERLEGO è nato con lo scopo di simulare i processi termoidraulici legati alla generazione di energia elettrica. Viene quindi molto spesso omessa la simulazione sia della sottostazione elettrica a partire dal generatore associato alle turbine sino alle linee in alta tensione in uscita dall'impianto. Vengono parimenti semplificati tutti i motori elettrici generano la potenza meccanica alle macchine operatrici che fanno parte dell'impianto simulato.

In effetti le dinamiche dei processi termoidraulici di un ciclo per la produzione di energia elettrica si possono considerare fortemente disaccoppiate dalle dinamiche dei fenomeni elettromeccanici (ed ancora di più da quelli elettromagnetici) relativi alla rete elettrica a cui è collegato l'impianto. Per questo motivo si può ritenerne accettabile il:

- trascurare i transitori sulla produzione di energia meccanica introdotti dalla regolazione di tensione
- trascurare le conseguenze che transitori di tensione sulle sbarre che alimentano le utenze dell'impianto (pompe, compressori, ecc) possono avere sul processo termoidraulico.

Qualora il simulatore debba contenere anche un modello di rete elettrica e sia richiesta la simulazione di transitori quali ed esempio il rifiuto di carico o il black startup è necessario rivedere la struttura delle catene di controllo relative principalmente le macchine operatrici sinora viste.

Bisogna per prima cosa ricordare che vista la potenza di calcolo richiesta dalla task del processo elettrico, normalmente ci si limita alla simulazione della parte del sistema elettrico comprendente:

- generatore elettrico
- sbarre MT e relative utenze
- trasformatore elevatore
- sbarre AT
- una porzione di rete nazionale sufficiente a fornire le condizioni a contorno necessarie alla simulazione dei transitori richiesti

Per le restanti parti (principalmente le sbarre BT con tutte le relativa utenze), si prevede normalmente un modello semplificato in grado di evidenziare gli effetti delle manovre sui load flow senza però riprodurre fedelmente i transitori elettromeccanici associati a tali manovre.

La struttura delle catene di comando degli attuatori sarà quindi differente a seconda che esista il modello fisico del motore elettrico di una macchina operatrice o che invece questo venga riprodotto in maniera semplificata.

5.1 Introduzione

Il motore elettrico può essere modellizzato in Legocad ricorrendo a due diverse soluzioni a seconda del grado di precisione desiderato. Se si intende costruire un modello del motore semplificato, si ricorrerà al modulo AMME (Figura 5.1), mentre, se si desidera una modellizzazione più accurata, ci si servirà del modulo MASI in coppia con il modulo GIPO (Figura 5.2).



Figura 5.1: Modulo AMME



Figura 5.2:Coppia dei moduli MASI e GIPO

5.2 CASO SEMPLICE: MOTORE ELETTRICO CON IL MODULO AMME

Il modulo AMME (ammettanza elettrica) simula gli assorbimenti elettrici del motore, il suo interruttore ed è fornito di un ingresso con il quale può essere interfacciato ad una pagina rego che implementa un relè di protezione elettrica; per ulteriori approfondimenti si rimanda alla documentazione del modulo.

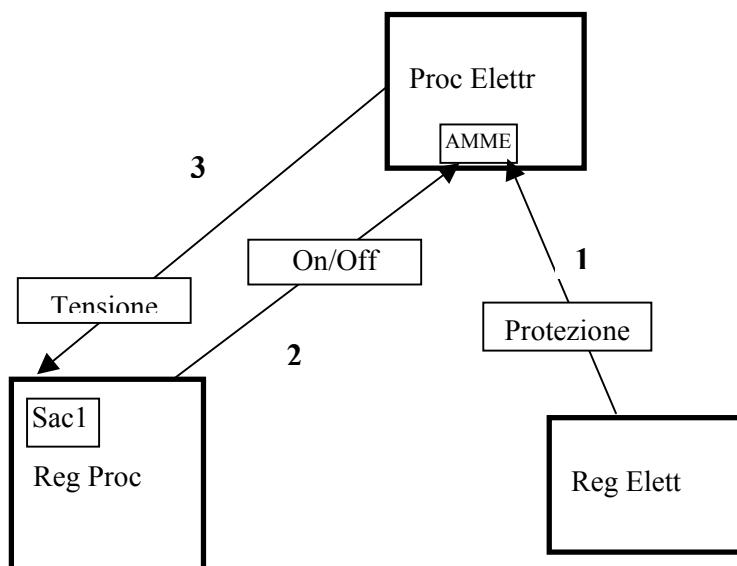


Figura 5.3: Connessioni del modulo AMME con le tasks di regolazione

La Figura 5.3 descrive i collegamenti tra il modulo AMME e le task di regolazione necessarie.

1. Come già accennato, dalle pagine di regolazione elettrica (Reg Elett), il segnale di scatto relè deve essere connesso all'ingresso SAMM del modulo, come mostrato in Figura 5.4: si noti che l'uscita del relè deve essere negata poiché il modulo AMME richiede per SAMM il segnale logico 0=Vero.
2. L'ingresso STAT (interruttore del motore) deve essere connesso alla pagina di regolazione che contiene l'attuatore della macchina operatrice a cui il motore elettrico fa riferimento, qui indicata con Reg Proc (solitamente ubicata all'interno della task di regolazione del sistema a cui il motore elettrico fa riferimento: nell'esempio riportato in figura, la task di regolazione dei fumi, poiché il motore in questione è un ventilatore).
3. La tensione di sbarra alla quale AMME è collegato deve essere connesso all'ingresso IV del modulo sac1 ubicato nella pagina di regolazione dove sta l'attuatore della macchina operatrice.

La Figura 5.5 mostra nel dettaglio una pagina di regolazione per un motore elettrico realizzato con il modulo AMME e i punti esatti in cui sono realizzati i collegamenti descritti in 2. e 3.

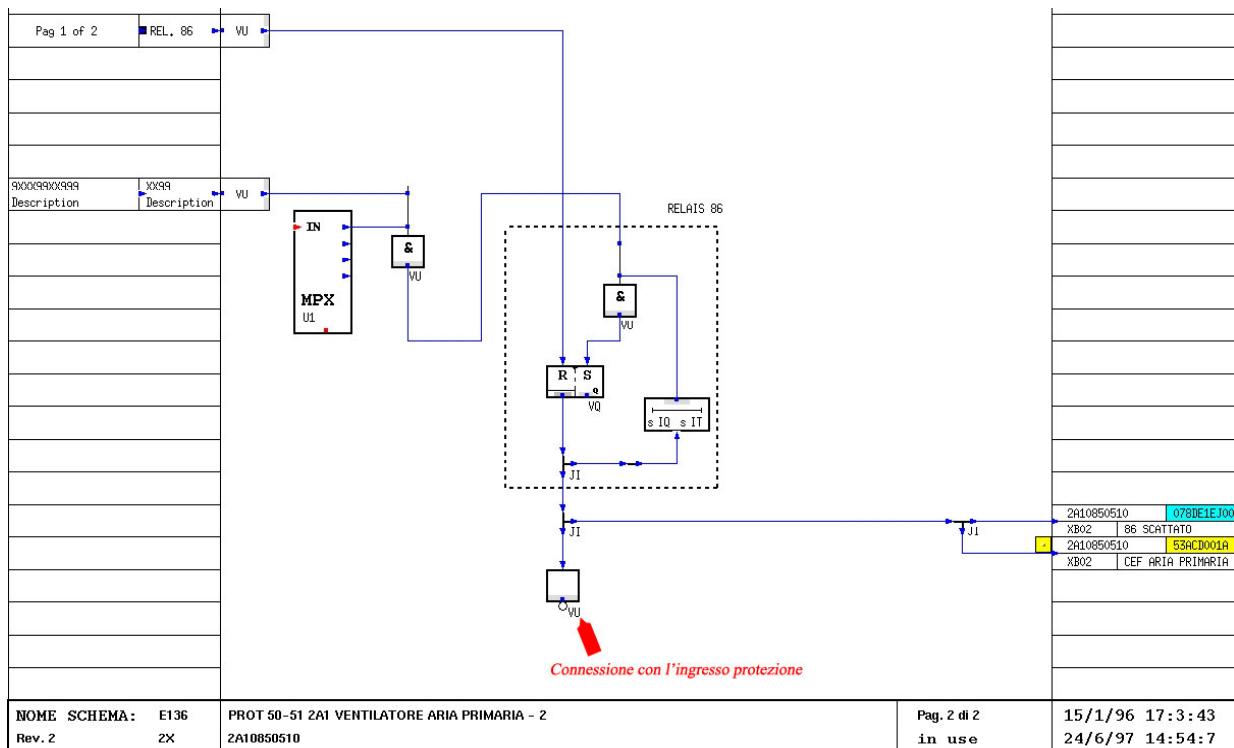


Figura 5.4: Collegamento con il relè di protezione elettrico

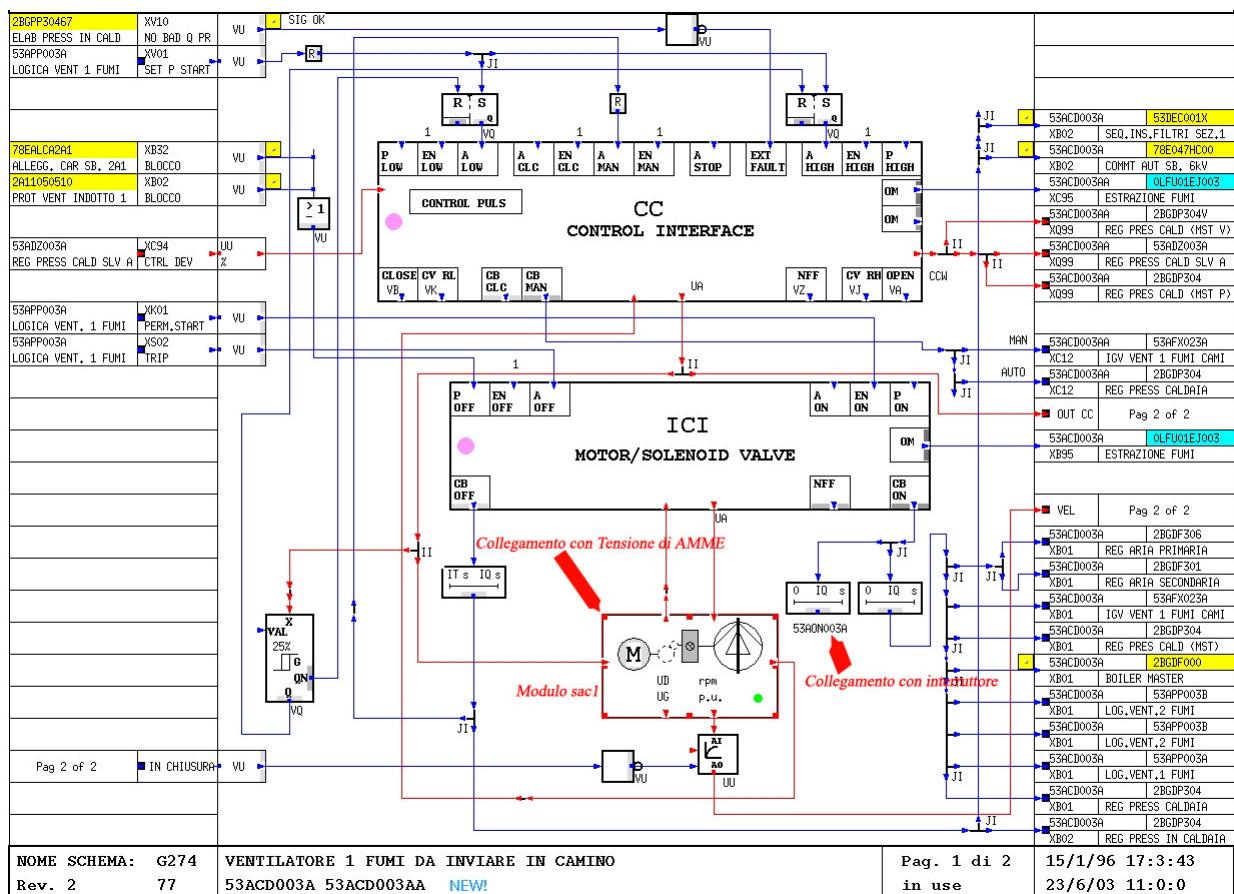


Figura 5.5: Collegamento con la pagina di regolazione della macchina operatrice

5.3 CASO “FULLY SIMULATED”: motore elettrico realizzato con la coppia masi-gipo

5.3.1 Generalità

Il modulo MASI simula dettagliatamente un motore asincrono e deve essere utilizzato in coppia con il modulo GIPO che serve da raccordo tra il motore elettrico e i moduli di processo, come per esempio CPOM (che modellizza la pompa); si rimanda alla relativa documentazione per i dettagli.

La connessione tra MASI e GIPO avviene tramite l'interscambio di due grandezze: MASI passa a GIPO la variabile QM00(=potenza meccanica) da connettere all'ingresso QASS(=potenza attiva fornita dal motore asincrono). A sua volta GIPO trasmette a MASI l'uscita OMMA(=velocità angolare del motore elettrico) da connettere all'ingresso OMMA(=pulsazione meccanica albero) (Figura 5.6).

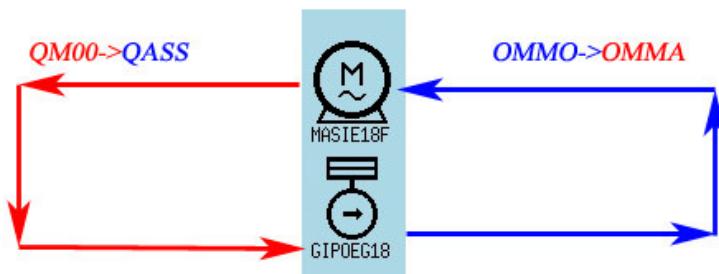


Figura 5.6: Connessione MASI-GIPO

5.3.2 Macchina operatrice a velocità fissa

Se la macchina operatrice a cui si riferisce il motore elettrico è a velocità fissa, nella parte dati del GIPO si deve aver cura di selezionare il TIPO=2 come indicato in Figura 5.7 e compilare la parte relativa ai “Dati della macchina operatrice” (si rimanda alla documentazione del GIPO per approfondimenti).

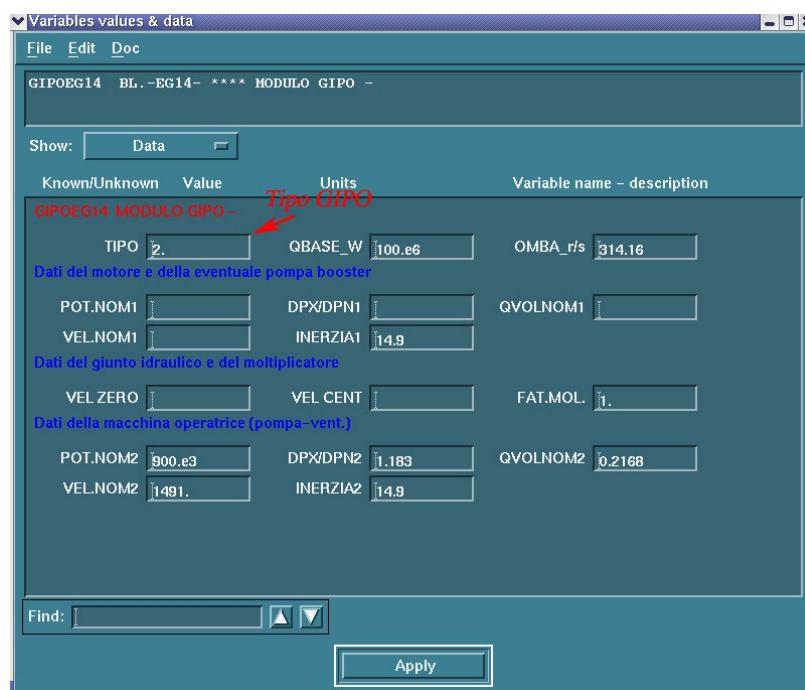


Figura 5.7: Interfaccia dati GIPO

La task di processo dove è ubicata la coppia MASI-GIPO deve essere connessa alla task di processo dove sta la macchina operatrice alla quale si riferisce (Proc Mecc in Figura 5.8) e alle task di regolazione dove si trova l'attuatore della macchina (Reg Proc) e quella in cui c'è il relè di protezione elettrica della stessa (Reg Elet).

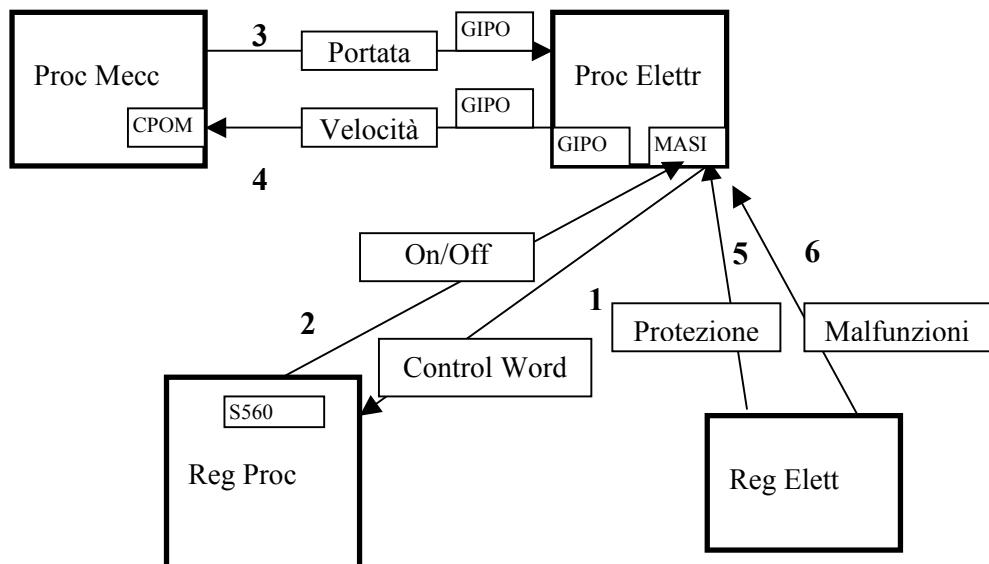


Figura 5.8: Connessioni MASI-GIPO con le altre task nel caso di macchina operatrice a velocità fissa

Si descrivono di seguito i collegamenti nel dettaglio:

1. MASI scambia l'uscita XCSO (=Controller Status Word) con l'ingresso IC del modulo s560 connesso all'attuatore della macchina operatrice.
2. L'uscita dell'attuatore, cioè il segnale di chiusura e apertura dell'interruttore, deve essere connesso all'ingresso STIN del MASI (segnale JI in Figura 5.9). Si noti che il segnale all'uscita dell'attuatore è normalmente =0 e solo in caso di chiusura o apertura varia rispettivamente di una quantità +1E-3, -1E-3 per un passo di tempo; per questo motivo si deve realizzare una struttura simile a quella mostrata in Figura 5.9 per fissare la variabile di stato interruttore in ingresso al MASI.
3. L'uscita OMPM (=Velocità angolare della macchina operatrice) deve essere connessa con quella della macchina (pompa/ventilatore/compressore) a cui si riferisce il motore elettrico.
4. GIPO riceve in ingresso dalla macchina operatrice di 3. la portata del fluido connessa alla variabile WPOM.
5. Il segnale di scatto del relè di protezione elettrica contenuto nella task Reg Elet deve essere connesso all'ingresso SMAS del MASI; si ricorda che, come già sottolineato nel capitolo 5.2, il segnale del relè deve essere negato (a tal proposito ci si riferisca a Figura 5.4).
6. Gli ingressi del MASI SMAL (=Malfunction del motore) e XMDE (=Malfunction Diagnostic Enable) devono essere connessi alle pagine di regolazione delle malfunzioni dei motori di cui se ne riporta un esempio in Figura 5.10 e che verranno discusse nel paragrafo 5.5.

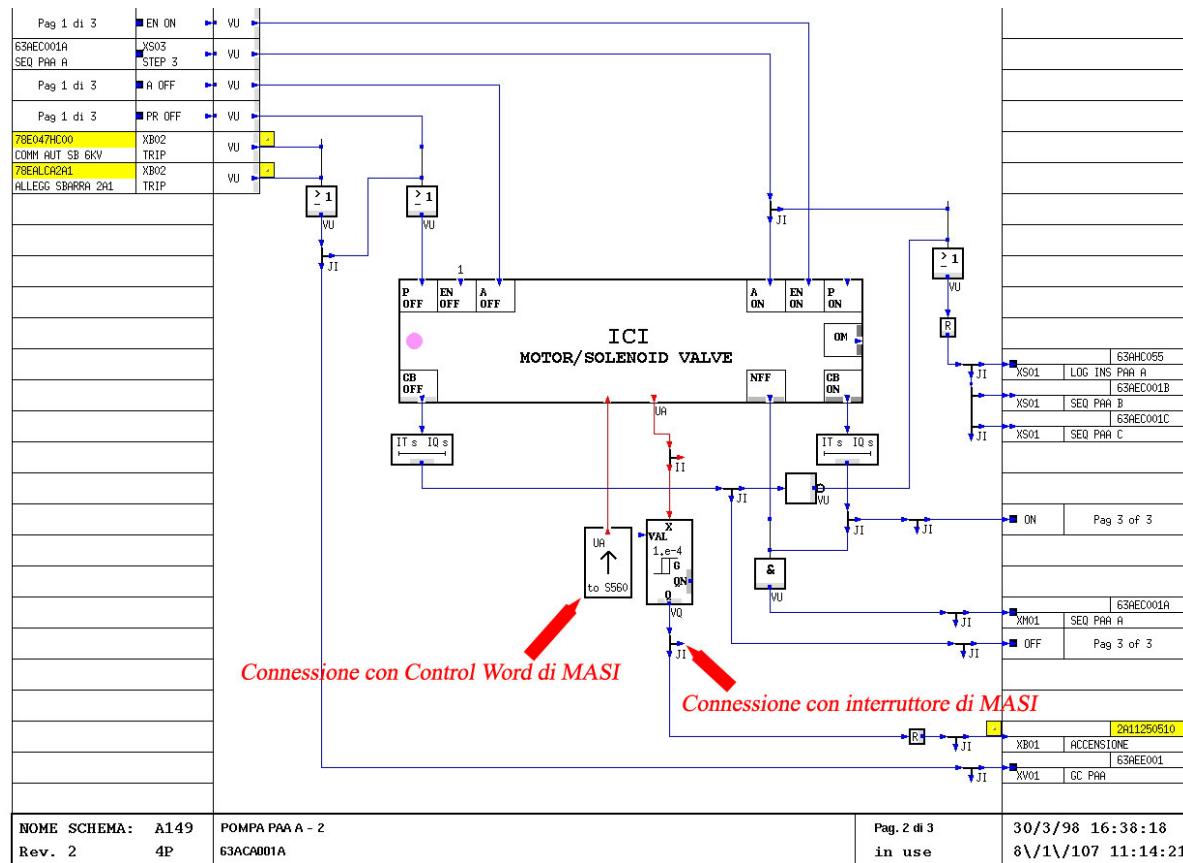


Figura 5.9; Connessioni tra Reg Proc e MASI

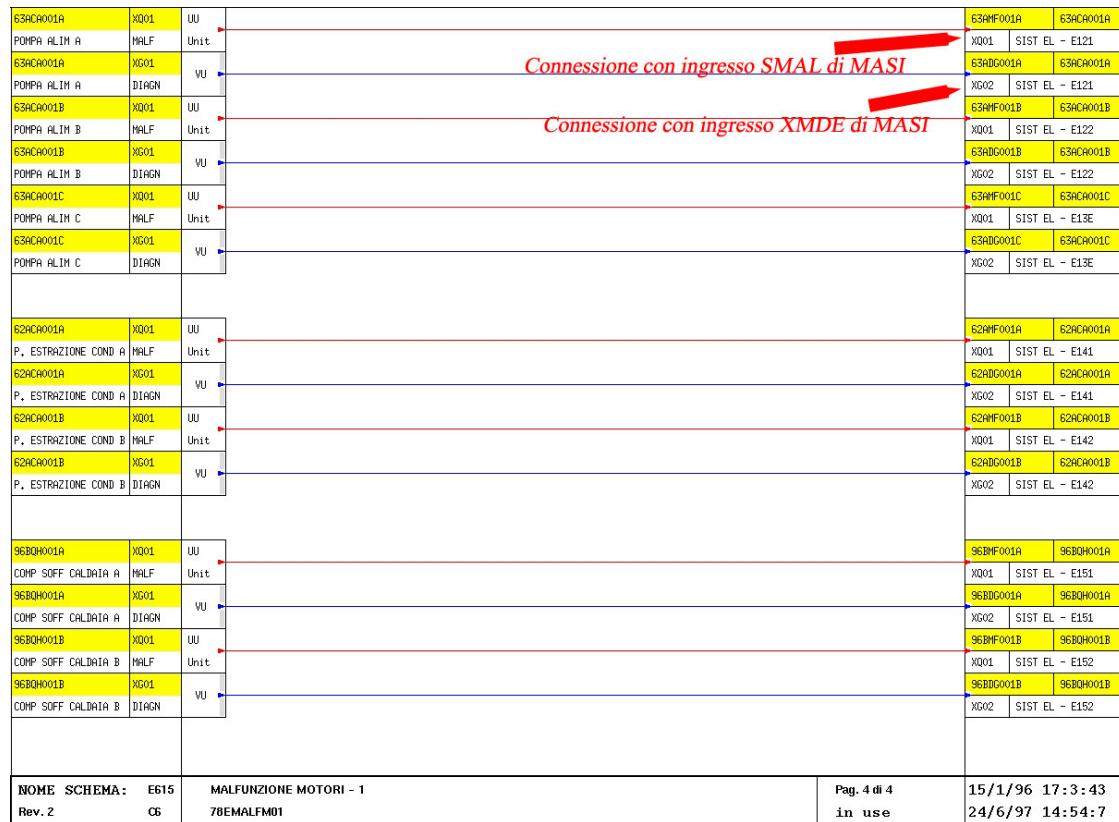


Figura 5.10: Connessioni tra la pagina di regolazione delle malfunzioni del motore elettrico e MASI

5.4 Macchina operatrice a velocità variabile

Nel caso la macchina operatrice modellizzata con la coppia MASI-GIPO sia a velocità variabile, nel foglio dati del GIPO si abbia cura di selezionare TIPO=1 e compilare le parti relative ai “Dati del motore e dell’eventuale pompa booster”, ai “Dati del giunto idraulico e del moltiplicatore” e ai “Dati della macchina operatrice” indicando la velocità minima e massima della macchina, come mostrato in Figura 5.11.

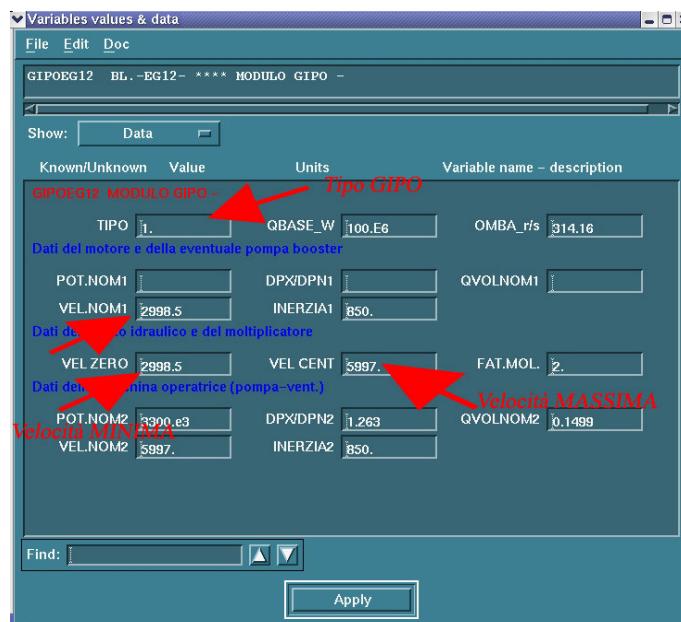


Figura 5.11: Foglio dati del GIPO per macchina operatrice a velocità variabile

Le connessioni tra i moduli MASI e GIPO con le altre task di processo e di regolazione rimangono le stesse con l’aggiunta di:

7. connessione dell’ingresso ALSC (=Posizione scoop) del GIPO con l’uscita UB del modulo sac3 di Reg Proc che rappresenta l’attuatore dello scoop (Figura 5.12 e Figura 5.13).

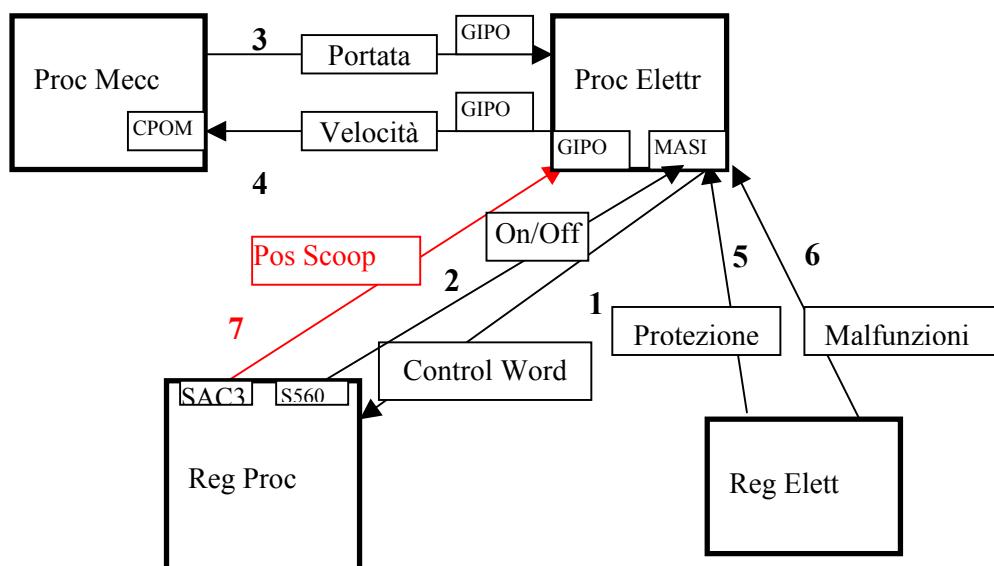


Figura 5.12: Connessioni tra MASI e GIPO e le diverse task

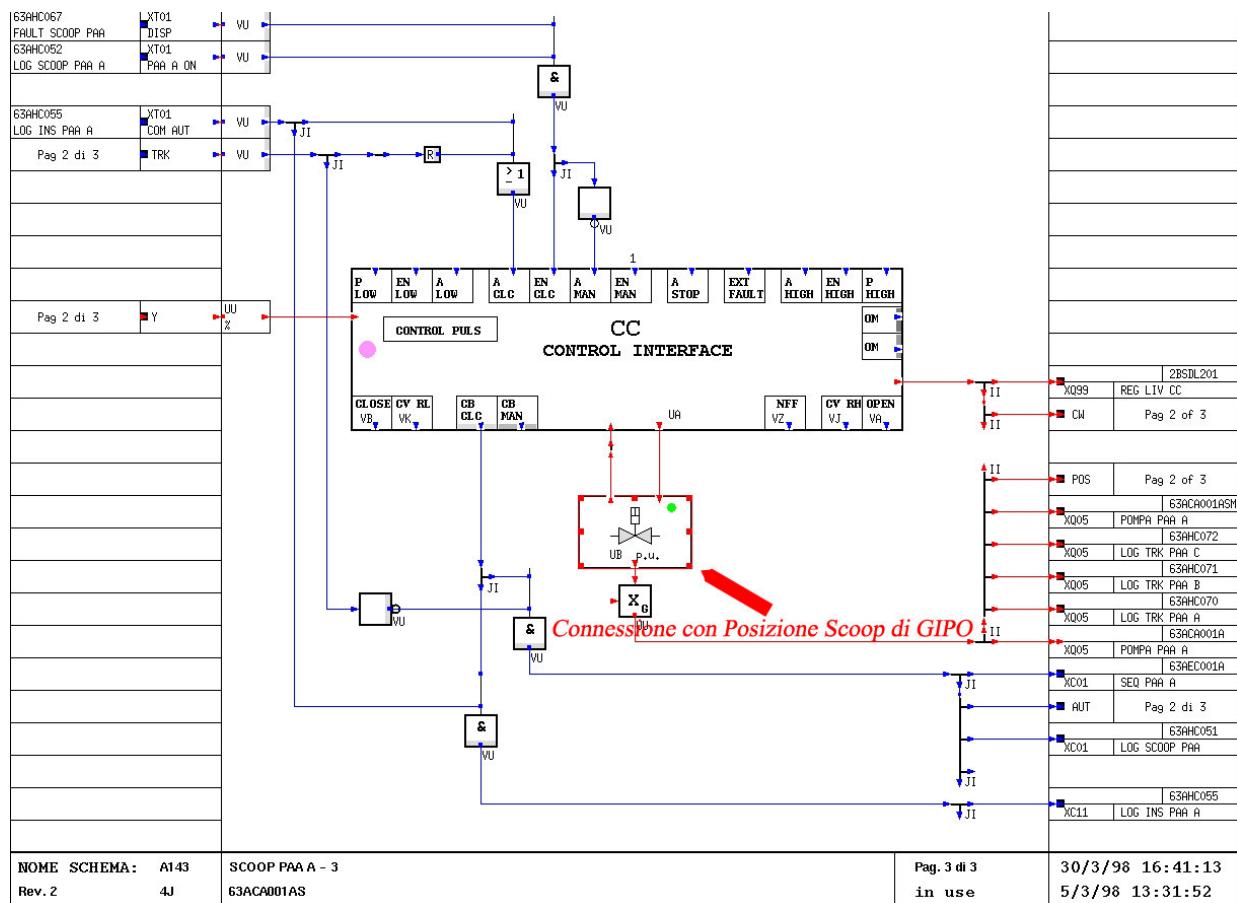


Figura 5.13: Connessione attuatore sac3 con posizione scoop di GIPO

5.5 Realizzazione del banco manovra per le malfunzioni del motore elettrico

Il banco manovra per le malfunzioni dei motori elettrici deve essere realizzato con pagine rego costruite come in Figura 5.14.

Per ogni motore elettrico si deve provvedere ad istanziare due ingressi del tipo “da altra task” uno analogico e l’altro digitale e le relative uscite da connettere ai masi come descritto nel paragrafo 5.3.2; nell’esempio seguente si fa riferimento al caso della Pompa Alimento della figura.

63HCA001A	XQ01	4Q		63AMF001A	63ACA001A
POMPA ALIM A	MALF	Unit		XQ01	SIST EL - E121
63HCA001A	XG01	4T		63ADG001A	63ACA001A
POMPA ALIM A	DIAGN			XG02	SIST EL - E121
63HCA001B	XQ01	4U		63AMF001B	63ACA001B
POMPA ALIM B	MALF	Unit		XQ01	SIST EL - E122
63HCA001B	XG01	4V		63ADG001B	63ACA001B
POMPA ALIM B	DIAGN			XG02	SIST EL - E122
63HCA001C	XQ01	4Y		63AMF001C	63ACA001C
POMPA ALIM C	MALF	Unit		XQ01	SIST EL - E13E
63HCA001C	XG01	4Z		63ADG001C	63ACA001C
POMPA ALIM C	DIAGN			XG02	SIST EL - E13E
62HCA001A	XQ01	52		62AMF001A	62ACA001A
P. ESTRAZIONE COND A	MALF	Unit		XQ01	SIST EL - E141
62HCA001A	XG01	53		62ADG001A	62ACA001A
P. ESTRAZIONE COND A	DIAGN			XG02	SIST EL - E141
62HCA001B	XQ01	56		62AMF001B	62ACA001B
P. ESTRAZIONE COND B	MALF	Unit		XQ01	SIST EL - E142
62HCA001B	XG01	57		62ADG001B	62ACA001B
P. ESTRAZIONE COND B	DIAGN			XG02	SIST EL - E142
96BOH001A	XQ01	5A		96BF001A	96BOH001A
COMP SOFF CALDAIA A	MALF	Unit		XQ01	SIST EL - E151
96BOH001A	XG01	5B		96BDG001A	96BOH001A
COMP SOFF CALDAIA A	DIAGN			XG02	SIST EL - E151
96BOH001B	XQ01	5E		96BF001B	96BOH001B
COMP SOFF CALDAIA B	MALF	Unit		XQ01	SIST EL - E152
96BOH001B	XG01	5F		96BDG001B	96BOH001B
COMP SOFF CALDAIA B	DIAGN			XG02	SIST EL - E152
NOME SCHEMA: E615			MALFUNZIONE MOTORI - 1		
Rev. 2	C6		78EMALFM01	KKS	
				Pag. 4 di 4 in use	15/1/96 17:3:43 24/6/97 14:54:7

Figura 5.14: Pagina rego delle malfunzioni dei motori elettrici

Nella tabella Component del DataBase si deve inserire una stringa costruita come segue:

ID CLASSE: **21**

DESCRIZIONE: **NOME DELLA MACCHINA OPERATRICE**

KKS: **KKS DELLA PAGINA DI REGOLAZIONE DOVE STANNO GLI INGRESSI DELLE MALFUNZIONI**

*Per i parametri seguenti si faccia riferimento all'esempio in figura;

COD REG VAR PRIMARIA: **4QC6EL-E615C6-ELET**

COD REG VAR SECONDARIA 1: **4TC6EL-E615C6-ELET**

COD REG VAR SECONDARIA 2:

COD REG VAR SECONDARIA 3:

COD REG VAR SECONDARIA 4:

TASK: **ELET**

@#L@TOPDES3: **NOME&DELLA&MACCHINA&OPERATRICE**

5.6 COME REALIZZARE IL RESET DEI RELE' ELETTRICI

Il reset dei relè elettrici è costituito dal modulo di regolazione MPX e da una struttura del tipo di quella mostrata in Figura 5.15

Nella tabella Component del DataBase si deve inserire una stringa come segue:

ID CLASSE: **24**

DESCRIZIONE: NOME DEL BANCO

KKS: KKS DELLA PAGINA DI REGOLAZIONE DOVE STA IL BLOCCO BANCO

*Per i parametri seguenti si faccia riferimento all'esempio in figura;

COD REG VAR PRIMARIA: **1D4GEL-E1934GEL-ELET**COD REG VAR SECONDARIA 1: **1C4GEL-E1934GEL-ELET**COD REG VAR SECONDARIA 2: **184GEL-E1934GEL-ELET**

COD REG VAR SECONDARIA 3:

COD REG VAR SECONDARIA 4:

TASK: ELET

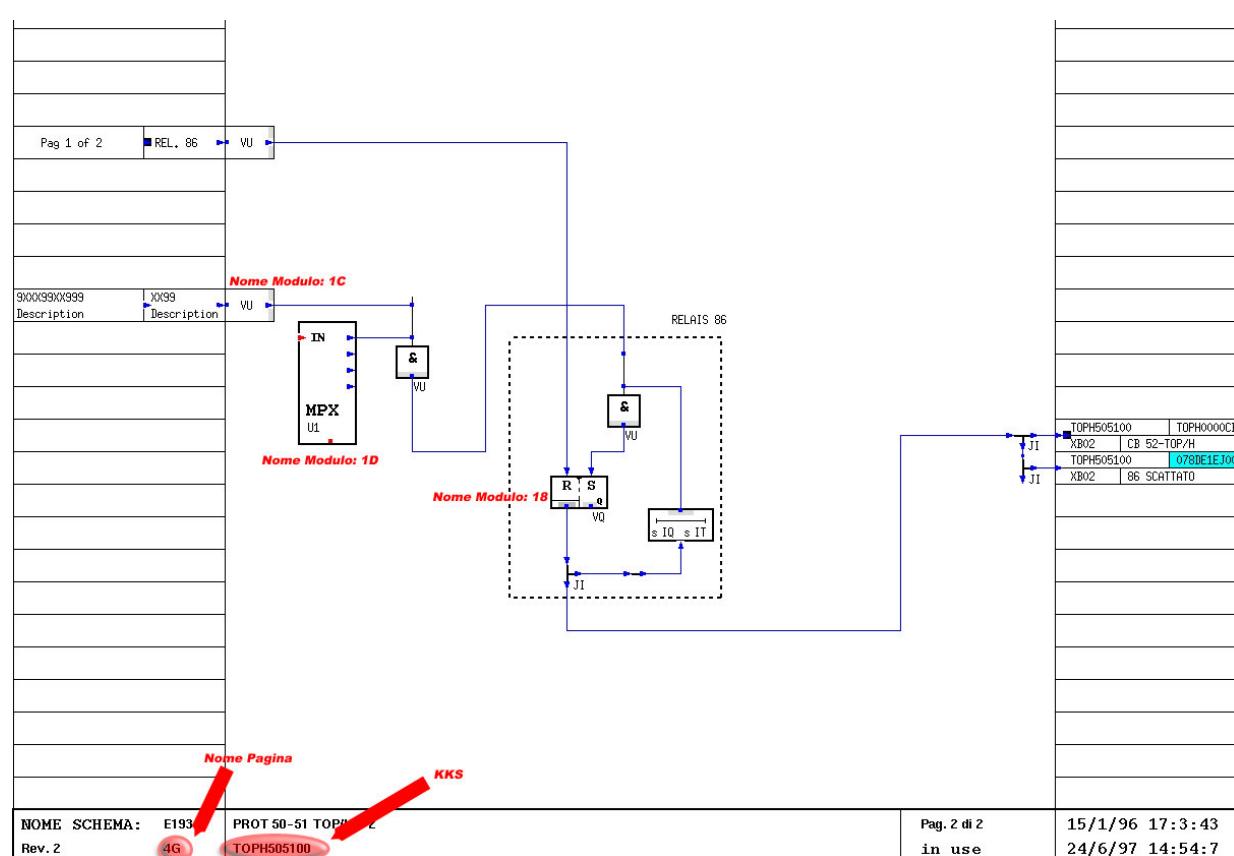
@#L@TOPDES3: **NOME&DEL&BANCO**

Figura 5.15

5.7 COME REALIZZARE IL “BANCO MANOVRA” DEGLI PSEUDOPROCESSI

Il “banco manovra” è costituito dal modulo di regolazione MPX e da una struttura del tipo di quella mostrata in Figura 5.16.

Nella tabella Component del DataBase si deve inserire una stringa come segue:

ID CLASSE: **16**

DESCRIZIONE: NOME DEL BANCO

KKS: KKS DELLA PAGINA DI REGOLAZIONE DOVE STA IL BLOCCO BANCO

*Per i parametri seguenti si faccia riferimento all'esempio in figura;

COD REG VAR PRIMARIA: **0K2PUT-X1502PUT-UTIL**COD REG VAR SECONDARIA 1: **0G2PUT-X1502PUT-UTIL**COD REG VAR SECONDARIA 2: **0B2PUT-X1502PUT-UTIL**

COD REG VAR SECONDARIA 3: **0H2PUT-X1502PUT-UTIL**

COD REG VAR SECONDARIA 4:

TASK: UTIL

@#L@TOPDES3: **NOME&DEL&BANCO**

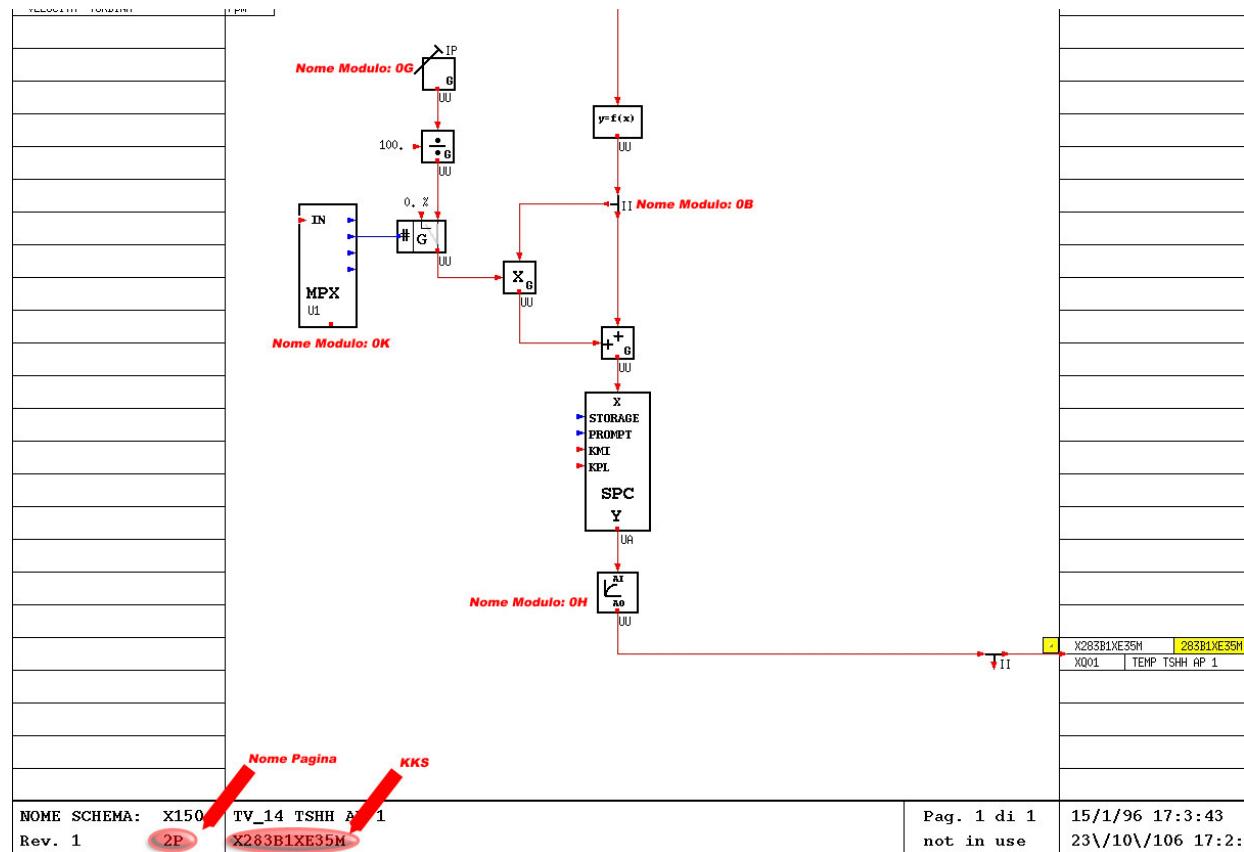


Figura 5.16

6 SISTEMA MMI

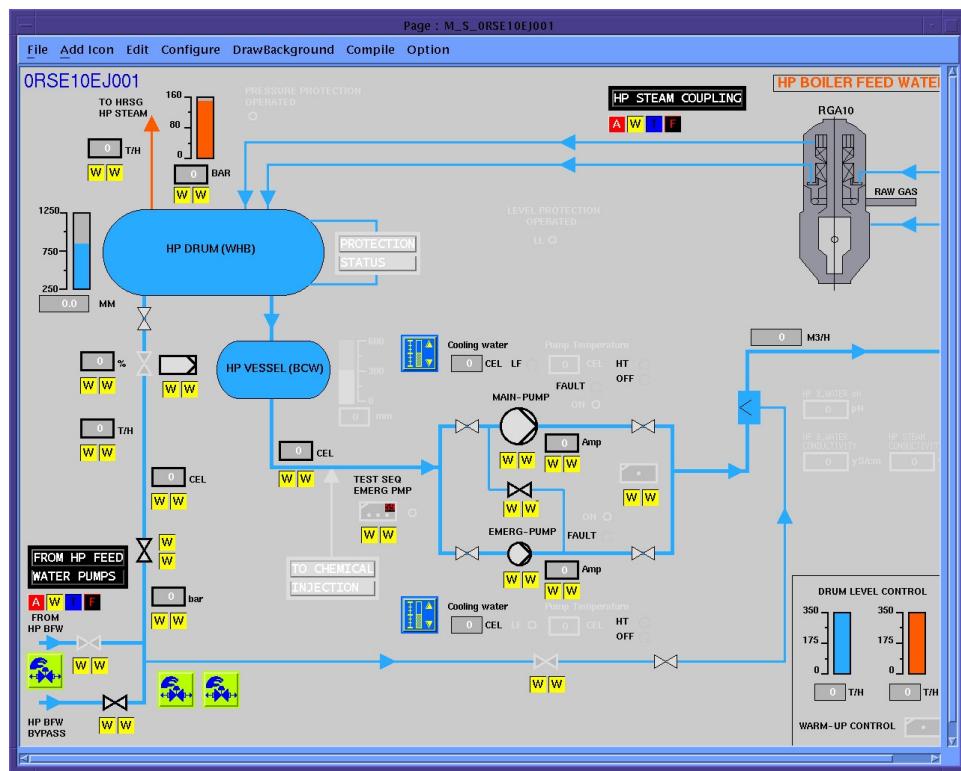
Il sistema di Man Machine Interface è costituito principalmente da quattro sistemi:

- pagine sinottico
- stazioni di controllo/supervisione
- sistema SCADA per gli allarmi
- trend variabili

6.1 Pagine MMI

La costruzione di pagine MMI è molto simile a quella delle pagine REGO. Chiaramente le librerie utilizzate conterranno elementi grafici, con programmi di animazione predefiniti anziché blocchi funzionali come nelle pagine REGO. Di conseguenza la configurazione di questi moduli si limiterà alla scrittura della pagina di stazioni collegata.

Come nelle pagine REGO ugualmente esiste un *drawer* per la realizzazione delle parti inanimate.



6.2 Stazioni di controllo/supervisione

Le stazioni di controllo (OW) servono per l'interazione tra operatore ed il processo e si esplica tramite il sistema di controllo precedentemente configurato nelle pagine REGO. Infatti ogni OW è intimamente connesso ad un determinato modulo nel sistema di controllo configurato nelle pagine REGO.

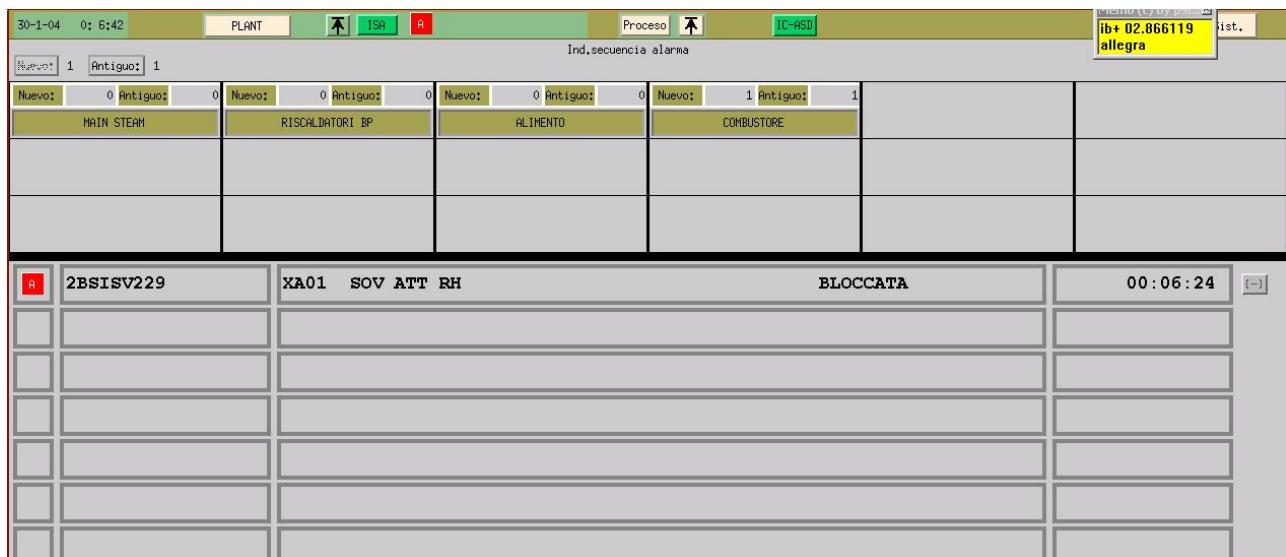
La configurazione delle OW è contenuta in un apposito file (*tipostazione.list*) definito per ogni tipo di stazione.

File di configurazione	Stazione	Blocco REGO associato
N_winDAU_2.list	DAU_2	Unit Switch-Over –2 system
N_winDAU_3.list	DAU_3	Unit Switch-Over – 3 system
N_winDBR.list	DBR	Set-Point Adjuster
N_winDEM2.list	DEM2	Pump
N_winDEM4.list	DEM4	Compressor
N_winDEM5.list	DEM5	Breaker
N_winDEM6.list	DEM6	Motor
N_winDES1.list	DES1	Valve
N_winDES3.list	DES3	Damper
N_winDEV1.list	DEV1	Solenoid Valve
N_winDEV3.list	DEV3	Solenoid Damper
N_winDFGK.list	DFGK	Sub Group Control - Master
N_winDFGK_S.list	DFGK_S	Sub Group Control - Step
N_winDFRC.list	DFRC	Continue Controller
N_winDFRS.list	DFRS	Step Controller
N_winDGS_2.list	DGS_2	Group Control
N_winDGS_3.list	DGS_3	Group Control
N_winDGWI.list	DGWI	Ing. Measure
N_winDGWI_P.list	DGWI_P	Ing. Measure (P)
N_winDGWM.list	DGWM	Perc. Measure
N_winDGWM_P.list	DGWM_P	Perc. Measure (P)
N_winDKR1.list	DKR1	Valve (Continue Controller)
N_winDKR3.list	DKR3	Damper (Continue Controller)
N_winDKR7.list	DKR7	Valve (Continue Controller)
N_winDSR1.list	DSR1	Valve (Step Controller)
N_winDSR3.list	DSR3	Damper (Step Controller)
N_winDTS.list	DTS	Sub Loop Control
N_winDVW_2.list	DVW_2	Preselect – 2 channel
N_winDVW_3.list	DVW_3	Preselect – 2 channel
N_winDVW_4.list	DVW_4	Preselect – 2 channel

Esiste inoltre un database tramite il quale è possibile generare i sopradetti files in maniera guidata.

6.3 Sistema SCADA

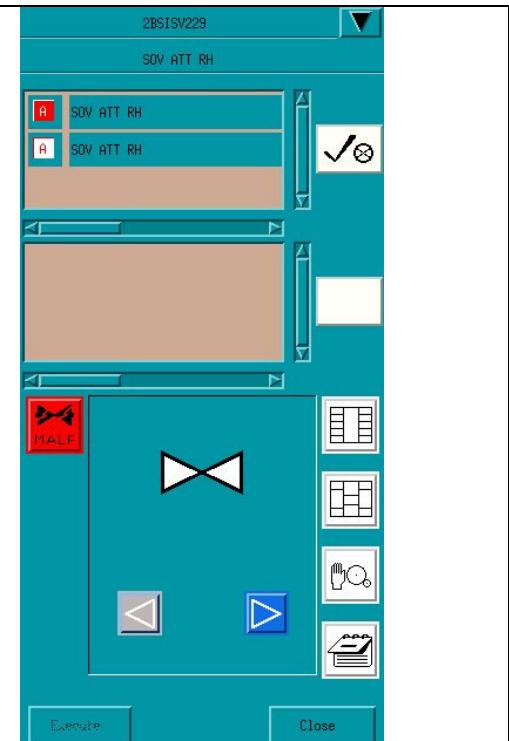
Si tratta di un sistema che consente la generazione e la visualizzazione di allarmi, sia in liste sequenziali, sia associandole ad oggetti di MMI.



La configurazione degli allarmi, alimentati da variabili generate nelle pagine REGO, viene fatta tramite un database in grado di generare i files necessari all'attivazione del sistema, che sono, principalmente:

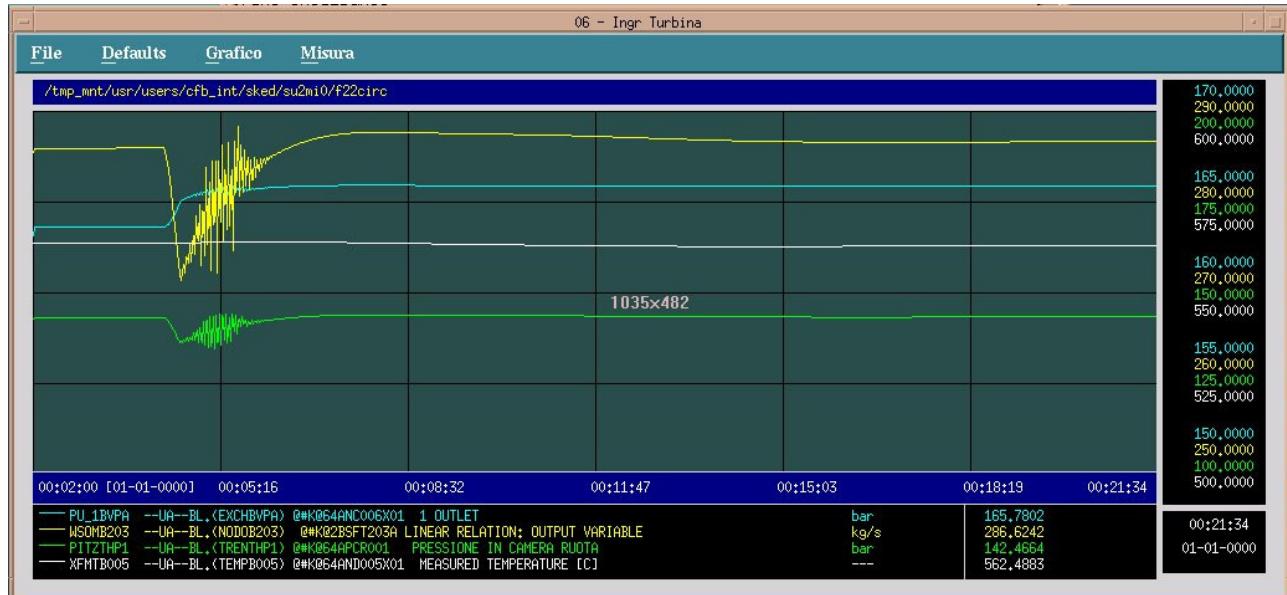
- ALARM.txt
- TAG.txt

Dai quali, attraverso apposite procedure viene generato il file fnomi.rtf che viene utilizzato durante le sessioni di lavoro.



6.4 Trend variabili

È il mezzo attraverso il quale è possibile visualizzare le variabili di processo/regolazione, a patto che siano state inserite, all'inizio della sessione di lavoro, nella lista di variabili da memorizzare.



7 CREAZIONE DI UN SIMULATORE

Vengono di seguito elencate le principali routine e gli interventi necessari per la generazione di un simulatore. La breve descrizione allegata presuppone l'utilizzo di tali procedure da persone già esperte dell'argomento.

1. Aggiornamento variabili d'ambiente

Aggiungere nel file *.profile* (ubicata nella directory utente) la variabile \$KSIMNAME=nome_simulatore
Creare la directory *nome_simulatore* sotto la directory sked

2. Verificare nella directory SIM, la presenza dei files

- ❖ al_sim.conf
- ❖ pd.list (Può essere generato con il comando kMakePdList)
- ❖ f22_fgraf.edf
- ❖ LeeF22.in
- ❖ recorder.edf
- ❖ page_mandb.dat

Possono essere generati tramite la macro **kArchiveLoad** che li richiama da precedenti versioni del simulatore. A questo punto sarà sufficiente apportare le modifiche legate al simulatore in costruzione.

3. **kCheckRegoTask Sim**

Verifica la presenza di tutti gli allarmi di sistema (tipo F) necessari per lo SCADA.

I possibili errori (registrati nei files .log nella directory log):

- Manca allarme di tipo F in un modulo che lo richiede
(W126.pag SETPOINT ADJUSTER: ERROR ALARM NOT CONFIGURATED)
- Lo stesso allarme è stato configurato due volte

4. Verificare che tutte le task di processo siano state correttamente compilate.

5. **kCompile Regolation Sim**

Partendo dal file *.pag* contenente la topologia ed i dati di ogni pagina REGO, genera, nella directory *proc* di ogni task, i files fortran equivalenti ad *f01.dat* ed *f14.dat* di ogni pagina.

6. **kCompile Task Sim**

Partendo dai file *f01.dat* ed *f14.dat* di ogni pagina REGO, genera, nella directory *proc* di ogni task, i files *f01.dat* ed *f14.dat* complessivo e quindi *lg5sk* (eseguibile), *variabili.edf* e *variabili.rtf* di ogni task.

7. **kConnex**

In funzione di quanto richiesto nel file di configurazione *al_sim.conf* crea il file **S01**, che contiene la configurazione del simulatore, comprese le connessioni tra task di processo e/o regolazione.

8. **kDiffS01**

Serve a verificare se esistono differenze sensibili (> 2%) tra i valori di variabili di regime tra i segnali connessi tra task diverse.

9. **kNetCompi**

Genera, nella directory del simulatore, il file **variabili.rtf** (che contiene la lista di tutte le variabili del simulatore), a partire dagli omonimi files di ogni task.

10. **kFindDuplicatedTag**

Verifica l'esistenza di uscite duplicate (con la stessa TAG).

11. kUpDateNavigation -i -Parent

Configura automaticamente le icone di salto pagina tra pagine REGO di task differenti.

12. kCompile Page Sim

A partire dai file *.pag* crea i files *.rtf* necessari per l'animazione tramite MMI.

13. kUpDatePD

Copia dall'utente *display* i PD's elencati nel *pd.list* del simulatore, nella directory *plant_display*

14. Creazione file di configurazione MMI e SCADA

Il data-base dell'automazione permette, se correttamente configurato, la creazione di tutti i files necessari per la generazione delle OW e dello SCADA. Utilizzando le apposite macro dovranno essere generati, e quindi trasferiti mediante FTP nella directory *export* del simulatore, almeno i seguenti files:

- ❖ *N_winTIPOOW.tab*: contiene i dati di configurazione delle OW, ne esiste uno per ogni tipo di OW. Verranno in seguito convertiti in *N_winTIPOOW.list*
- ❖ *Component.mf* | contengono tutte le informazioni per la creazione delle stazioni di malfunzione e per quelle dei Remote Parameter, necessarie per la gestione delle condizioni al contorno del simulatore
- ❖ *malf_set.mf*: |
- ❖ *tipo_comp_malf.mf* |
- ❖ *kpd.txt* | contiene la lista dei PD e le informazioni sulla relativa gerarchia
- ❖ *Alarm.txt* | contengono tutte le informazioni sugli allarmi necessarie per la generazione del sistema SCADA
- ❖ *Tag.txt* |
- ❖ *Tags.txt* |
- ❖ *cai_var.dat* | contiene le informazioni per la configurazione automatica dei cai_var
- ❖ *sosti.dat* | contiene le informazioni per l'assegnazione corretta delle gerarchie delle OW

15. kExport_new

Copia tutti i files presenti nella directory *export*, eseguendo dei controlli quando necessario, nella directory dove verrà utilizzato.

16. kCheckAlarm_new

Esegue molte verifiche di ausilio al sistema SCADA e principalmente:

- a) Selezione solo gli elementi delle aree funzionali dichiarate nel simulatore (nel *al_sim.conf*: FA 1 2 3 4 5 7 11 12 13 ... 18). Viene verificata l'associazione degli allarmi ad aree funzionali comprese nel simulatore; gli altri allarmi non vengono utilizzati
- b) Verifica che tutti gli allarmi siano associati ad una OW o ad un PD della medesima FA
- c) Verifica che tutte le OW abbiano una gerarchia
- d) Selezione tra tutte le OW contenute nei files *N_winXXX.list*, solo quelle di FA appartenenti al simulatore

17. kDirect

Copia i PD nella directory *o_win*, eseguendo al contempo la sostituzione tra KKS e varibili LEGO

Nota1: questa Macro cancella tutti i PD precedentemente presenti nella directory

Nota2: se si vuole elaborare un solo PD procedere come di seguito descritto:

- ❖ *kbasic*
- ❖ *direct.sh -s igcc_1 -p KKSPD*
- ❖ *mye direct.sh.log* (per verificare i messaggi d'errore)

Segue la descrizione degli errori più frequenti:

- a) quando il file *.log* non chiarisce l'errore procedere come segue:
 - *kbasic*
 - *chgvar < M_S_NOMEFILE.pag > M_S_NOMEFILE.pag.new*
 - *\$mye error_chgvar.msg*
- b) gli errori nei link diretti si chiariscono sempre nel file *.log*
- c) se alcune OW's non sono presenti nel file *superlista.list*, la causa può essere:
 - non è correttamente configurato il data-base

- non è stato correttamente trasferito il file di configurazione nell'ambiente LINUX
 Gli errori di tipo a e b devono essere corretti nei PD's originali, cioè nell'utente display.
 Una volta corretti gli errori eseguire nuovamente i comandi KupDatePD e kDirect. Qualora sia stata modificato il data-base, occorre prima reimporarlo ed eseguire il kExport.

18. kOw -nodetail

Genera le OW e le Detail Windows sotto o_win partendo dai *N_winXXX.list* corrispondenti, creando .pag e .bkg. Qualora sia richiesta la creazione delle DETAIL WINDOW omettere l'opzione “-nodetail”

Nota: questa Macro cancella tutte le OW's precedentemente presenti nella directory

19. kMakeCurve Check

Questa Macro provvede a:

- Verificare che tutte le variabili associate ad un grafico siano configurate (nella tabella Variables de graficación y Grupos de Graficos nella sezione Menu Graficos Simulador del data-base)
- Genera i PD's di curve sotto o_win (insieme agli altri PD's, OW, etc)
- Verifica che le variabili selezionate per ogni gruppo di grafici esistano nel simulatore.

Errori: gli errori sono riportati nei files .log e .err (in entrambi appaiono gli stessi errori, ma nel .err è incluso il KKS, e nel .log il tipo di errore: *lego variable UUOUTACO not found in variabili.edf*, in questo caso la variabile non è presente nel simulatore. Va corretto quindi il data-base.

Nota: questa Macro cancella tutti i PD's di grafici precedentemente presenti nella directory

20. kScadaInit

Compila i file SCADA e genera il file *fnomi.rtf*

Nella directory del simulatore si trova la directory *scada* che contiene: *PointInst.err*, *fileop*, *rtf*, *star_txt* e *txt*. Nella directory *start_txt* si trovano i files esportati *ALARM.txt* (che include la gerarchia) e *TAG.txt* (include i KKS's), tra l'altro. Quando si lancia la macro kScadaInit viene subito creata nella directory *txt* una traduzione binaria di questi files, dopo vengono creati i *.rtf* nella directory *rtf* e, infine, viene creato il file *fnomi.rtf* (che contiene la lista di tutti gli eventi del simulatore) e *conf.cfg* nella directory *fileop*. Il file *fnomi.rtf* sarà utilizzato dalla Macro kCompileSim (perché vengano compilati anche gli allarmi è necessario che la variabile d'ambiente SCADA_DEBUG = 0 che può essere verificato col comando “env | grep SCADA”); il file *conf.cfg* contiene l'indirizzo del fileop; inoltre si crea un link del file *fnomi.rtf* nella directory del simulatore

NOTA: per cercare un KKS nel file .rtf si può lanciare il comando:

```
$strings fnomi.rtf | grep HAD10
```

21. kMakeConnDB

Genera un data-base da usare durante la compilazione delle Detail windows.

Il risultato è il file REG_INT_CONN_DB presente sotto o_win. Può essere letto (per la ricerca di variabili connesse) con il comando: \$kReadDB REG_INT_CONN_DB *variabile*

Va utilizzato solo qualora sia stata precedentemente prevista la realizzazione delle Detail Window

22. kUpDateCaiHierarchy

Questa Macro esegue le seguenti funzioni:

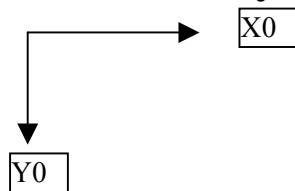
- Scrive la corretta gerarchia al posto della stringa de KKSOW_X in tutti i CAI
- Configura i 4 CAI's associati ad ogni PD
- Assegna ad ogni PD la gerarchia corrispondente in funzione di quella assegnata nel data-base

I principali tipi di errore sono i seguenti:

- a) *Hierarchy of OW 0RPJ77CP001_DGWM in PD M_S_0RAH10EJ001.pag not found in sosti.dat*. Questo errore indica che non si trova la gerarchia nel file *sosti.dat*, questo non è un problema di questo file ma del

- nome con cui è configurata la OW nel PD che è errata; in questo caso manca una P:
0RPJ77CP001_DGWM_P
- b) Errori trascinati da altre macro che non sono stati risolti nell'assegnazione delle gerarchie

Per localizzare il CAI, ricordare che le coordinate sono riferite agli assi indicati in figura



23. kMakeCaiVar

Inserisce tutti gli allarmi con la medesima gerarchia nel CAI corrispondente

Ogni CAI può contenere un massimo di 10 allarmi..

Errori: se ad un CAI sono associati più di 10 allarmi appare un WARNING. Se è un CAI di un PD si può tralasciare di intervenire; viceversa se si tratta di un CAI associato ad una OW questo deve essere configurato manualmente.

24. kChangeMmiColor

Cambia i colori del MMI per adeguarli allo standard TELEPERM

25. kPlace

Genera i .pag delle stazioni di Malfunzione e dei Remote Parameter

Nota: questa Macro cancella tutte le Malfunzioni ed i RP precedentemente presenti nella directory

26. kCheckSimulator

Esegue alcuni controlli formali sui PD

Errori più frequenti:

- a) *Malfunction of OW [O_0RCG40CS001_DGWM.pag] not found.* Indica che non è stata configurata nel data-base la stazione di malfunzione segnalata o un errore nell'assegnazione del nome alla malfunzione
- b) *The hyperlink [F_0LBA55AA002] of th PD_... not found.* Indica la mancanza nel data-base del RP segnalato o un errore nell'assegnazione del nome al RP

27. kWinContext

Prepara il *Context.ctx* di tutto ciò che si trova sotto *o_win*: PD,O.W., F_, ..

27. Controllo della gerarchie MMI

Dal **config** sotto *o_win* lanciare il comando check hierarchy dal menù file

Verifica la correttezza della gerarchia dei PD's e degli allarmi eseguendo un controllo incrociato con la gerarchia delle OW.s

Controllare il file OITree.log prodotto sotto *o_win*, dopo avere eliminato i report privi di significato con il comando

```
grep -v "ALARM OK" OITree.log | grep -v _DTS | grep -v _DVW | sort | uniq>errori.txt
```

Gli errori più frequenti sono dovuti a:

- Tutti i PD's devono avere almeno un allarme associato, se effettivamente non esiste crearne uno fittizio nel data-base e in una pagina REGO denominato ALWAYSFALSE_XX00
- Mancanza di allarmi di tipo F
- Tutte le OW devono avere almeno un allarme associato (generalmente almenos quello di tipo F), ad eccezione delle OW tipo _DTS e _DVW
- OW associate a PD's cancellati
- OW's con la stessa gerarchia

28. kCompileSim

Compila tutti i .pag presenti sotto *o_win*, generando i .rtf

In caso di errori ricompilare l'oggetto tramite il config verificando, dal menù Compile, i messaggi contenuti in Page Error View

29. kModRrtf sca/sim

Modifica i file *.rtf*, per adattarli al sistema SCADA, permettendo così il riconoscimento degli allarmi nei PD's. Qualora non si voglia utilizzare il sistema SCADA utilizzare l'opzione *sim*

30. kCollect

Unisce tutti i compilati (pagine REGO, OW e PD). Inoltre crea il file *kMmi.cfg*, che indica il tipo di MMI attivabili da differenti postazioni (sotto il simulatore)

Qualora sia necessario modificare il *kMmi.cfg*, può essere fatto manualmente rendendo operativa la nuova configurazione successivamente lanciando il comando *kMmiConfig* in ogni macchina.

Esempio (la struttura del file *kMmi.cfg* prevede: opzione; n° corrispondente; tipo di MMI (I: istruttore , O: operatore, S: solo SCADA); nome della macchina (elco0, elco1, etc); nome dell'utente (igccr1, oper1, oper2, etc):

```
a;1;I;elco0;igccr1
b;2;O;elco1;oper1
c;3;O;elco2;oper2
d;4;S;elco3;scada
```

Si possono avviare sino ad un massimo di 4 MMI contemporaneamente.

31.kMakeRecorder

Raccoglie tutte le variabili presenti nei files che seguono nel file *recorder.edf*.

- il vecchio *recorder.edf*
- il file importato dal data-base (tabella Variables de graficación y Grupos de Graficos)
- *f22.fgraf.edf* (che contiene le variabili presenti nei gruppi di grafics)

32. kFindKks

Permette l'uso del comando KKS per la ricerca dei KKS

33. kPulirSim (kPulirRegoAll)

Cancella tutti i files inutili nella directory legocad e sked.

34. kScd

Lancia l'applicazione SCADA

36. kStart

Lancia l'applicazione *net_startup* che gestisce la simulazione

37. kMmi

Lancia l'applicazione MMI di interfaccia operatore

8 DOCUMENTAZIONE COMPLEMENTARE

La documentazione di dettaglio sulla configurazione e l'utilizzo dei tools finora descritti è disponibile in formato informatico nei seguenti files:

aldb_tablestruct.doc	Alterlego Database – Struttura Tabelle
aldb_techspec.doc	Alterlego Database – Technical Specification
Aut_FDOWtest.doc	Automation F.D. s. & O.W.s. testing guidelines
Automation_and_mmi_func_spec_and_implem_guideline.doc	AUTOMATION & M.M.I. Functional Specification & Implementation GUIDE LINES

<u>MMI_FES4.DOC</u>	SIMULATOR MMI FEATURES
<u>mmi_fs_imp_guide.doc</u>	AUTOMATION & M.M.I. Functional Specification & Implementation - GUIDE LINES
<u>RUN_TIME_UUMAN.doc</u>	RUN TIME USER MANUAL
<u>scd_ifx_db_integr.doc</u>	SCADA-OSF1 Informix database integration “Technical Specification
<u>scd_prod_descr.doc</u>	SCADA-OSF1 Supervisory Control and data Acquisition “Product description
<u>scd_usermanual.doc</u>	SCADA-OSF1 Scada “User Manual
<u>Sim_stru_over.doc</u>	SIMULATOR STRUCTURE OVERVIEW
<u>simulator configuration procedure.doc</u>	SIMULATOR CONFIGURATION PROCEDURE

9 DATA-BASE DELL'AUTOMAZIONE

Il database dell'automazione contiene tutte le informazioni per la creazione del sistema MMI e del sistema SCADA del simulatore ed è realizzato con il programma Microsoft ACCESS.

Contiene inoltre tutte le informazioni sulla documentazione utilizzata nella realizzazione del sistema di controllo e le relazioni tra gli schemi di impianto e quelli effettivamente realizzati nel simulatore.

La comprensione di quanto segue, specialmente per quanto riguarda la parte relativa alla descrizione dell'architettura dell'applicativo, richiedono una sufficiente conoscenza del programma ACCESS.

Il database è stato progettato per gestire tutti i componenti dell'impianto, e quindi del simulatore, utilizzando la codifica KKS. Questa è in grado di individuare univocamente qualsiasi elemento costitutivo l'impianto.

Il formato della stringa è di norma il seguente:

0 TEC 52 CP 106 A

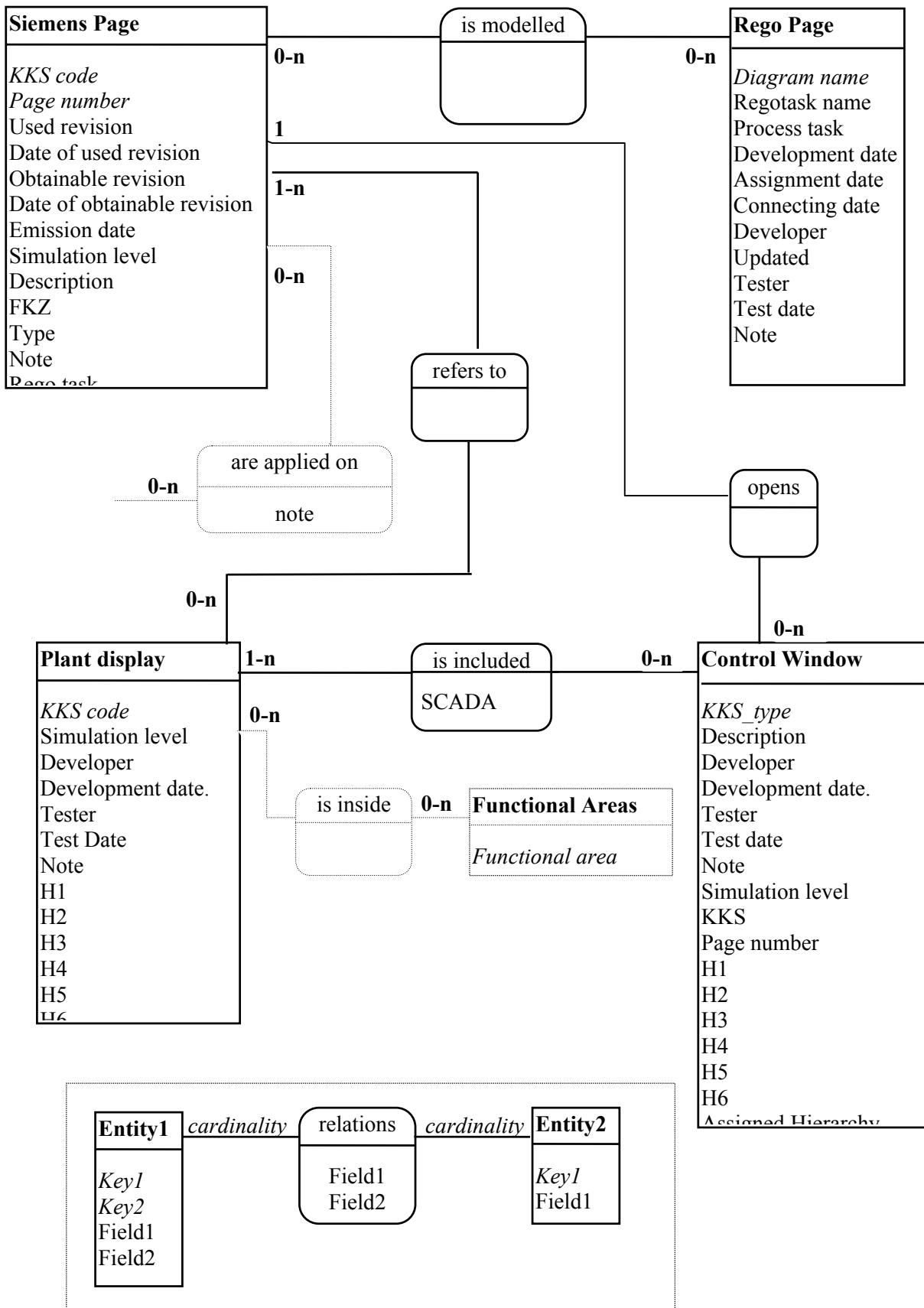
- ▶ progressivo di elemento multiplo, ad esempio nel caso di misure doppie o triple
- progressivo di elemento
- classifica elemento (es. CP= misura di pressione, AA= valvola, FF= logica, ecc)
- progressivo di sistema
- classifica sistema (es. TEC= produzione O2, MBP =alimentazione metano TG, ecc)
- progressivo impianto (es. 0 =sist comuni, 1 =turbina a gas, 2 =turbina a vapore)

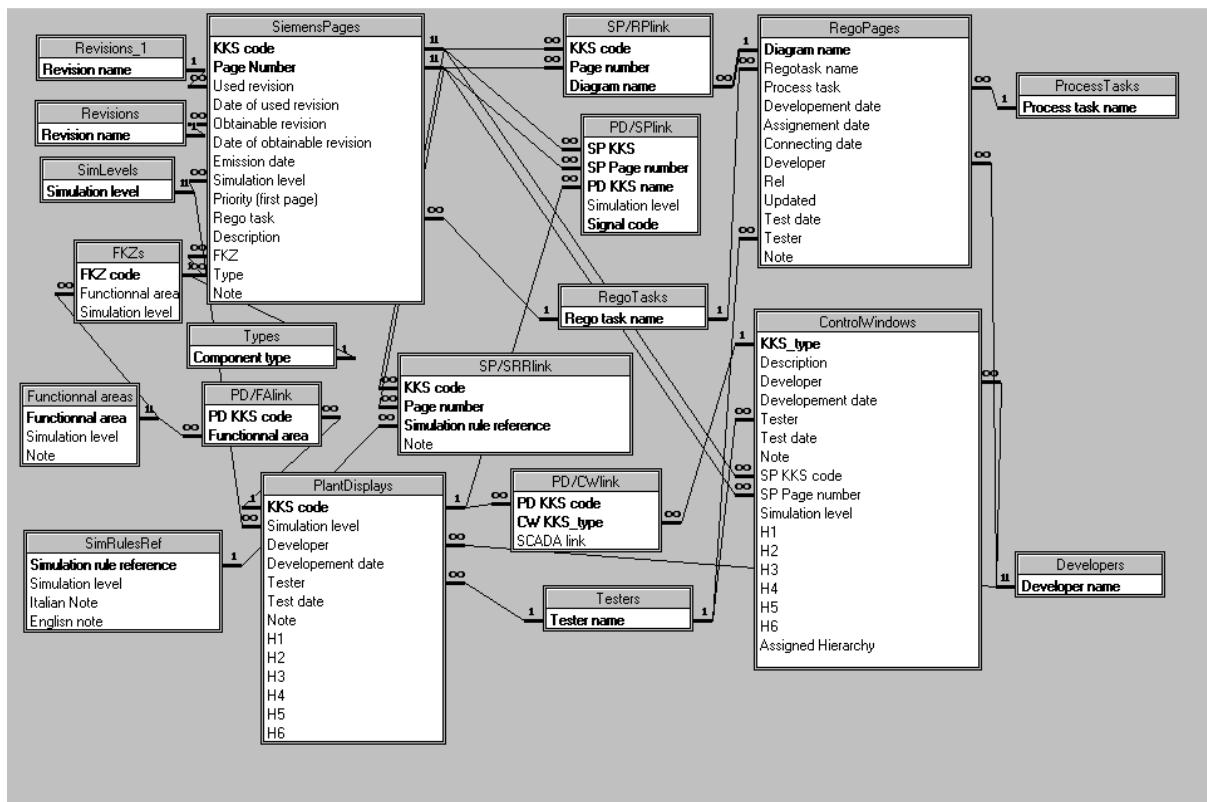
9.1 ARCHITETTURA

Il database può essere utilizzato contemporaneamente da più persone quindi il file contenente i dati e quello contenente l'applicazione sono stati separati in modo da evitare conflitti di rete.

- Il file ACCESS 7.0 afsdb4.mdb contiene le tabelle e la definizione delle relazioni tra le tabelle stesse. Solo l'amministratore del sistema dovrebbe potere modificare la struttura di quest'applicativo.
- Ogni utente disporrà del file ACCESS 7.0 afsin7.mdb contenente tutti i moduli, le queries, i reports e le maschere dell'applicazione. L'utente lavorerà quindi direttamente su questo. Naturalmente esistono dei link con il precedente database dove sono immagazzinati i dati. Ogni utente potrà personalizzare la propria applicazione con l'aggiunta di nuove maschere, queries, reports senza il pericolo di corrompere i dati comuni che sono soggetti al controllo delle regole nell'applicazione centrale.
- Il link tra i due database è realizzato tramite il protocollo ODBC
- Due diversi utenti possono lavorare contemporaneamente sullo stesso record. In questo caso un messaggio di warning sarà generato verso entrambi con la richiesta del record da memorizzare.

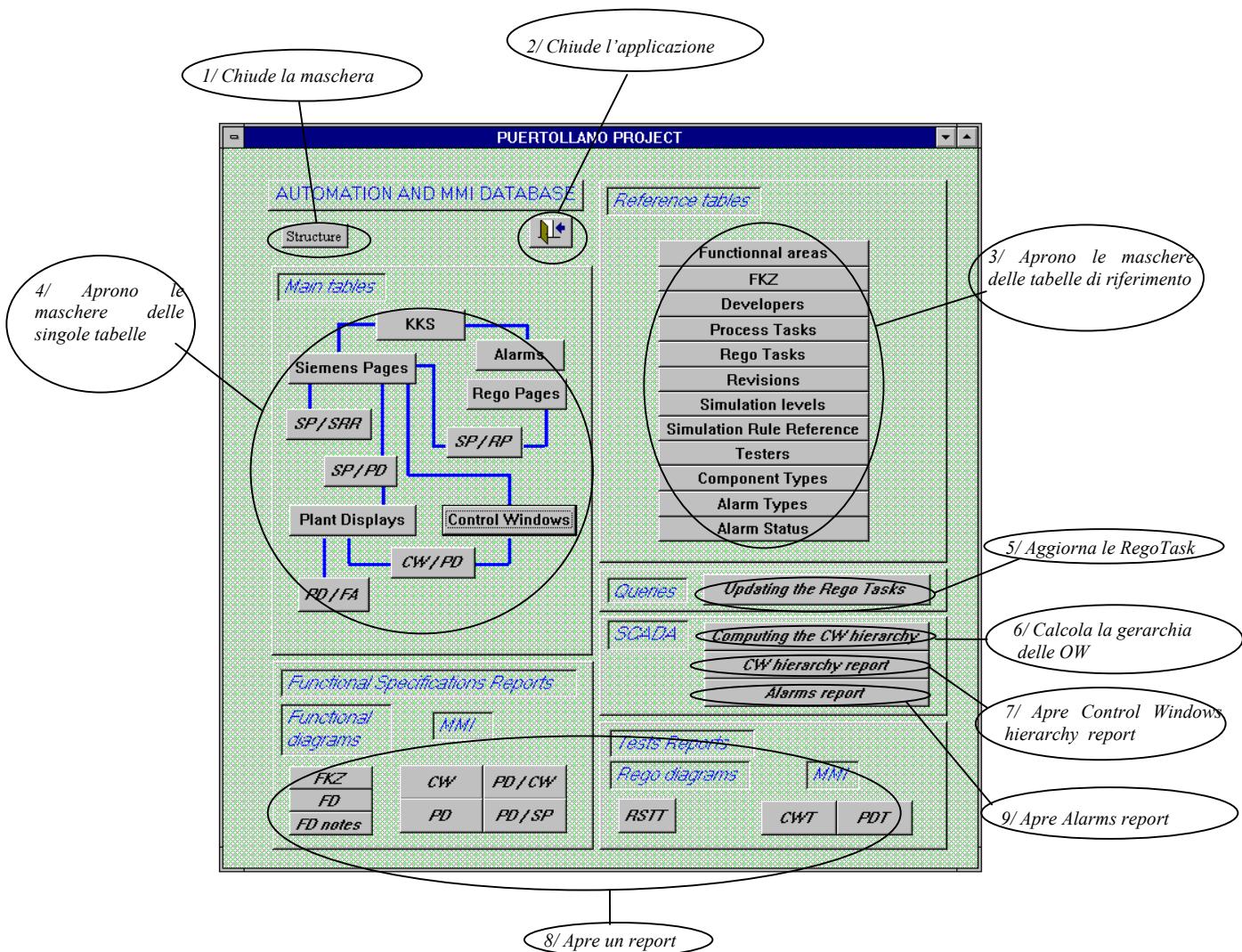
Seguono i diagrammi con la descrizione sommaria della struttura del database.





9.2 Descrizione dell'interfaccia utente

All'apertura del file "afsin6.mdb" comparirà la seguente maschera



1/ Chiude la maschera principale, senza uscire da ACCESS.

2/ Chiude la maschera ed esce da ACCESS.

3/ Ognuno di questi tasti permette l'accesso ad una maschera delle tabelle di riferimento

4/ Ognuno di questi tasti permette l'accesso ad una maschera delle tabelle dati. Le linee blu indicano i link logici tra le diverse tabelle.

5/ Questo tasto permette, previa selezione della REGO TASK richiesta, l'aggiornamento delle versioni della documentazione presente nella tabella SIEMENS PAGE dal campo OBTAINABLE REVISION a quello USED REVISION per tutte le REGO PAGES segnate come UPDATED.

6/ Questo tasto avvia una macro che calcola la gerarchia delle CWS aggiornando i campi H1,...H6 nella tabella CONTROL WINDOWS. Seguono ulteriori informazioni sulla gestione della gerarchia delle CWS.

7/ Questo tasto genera un report contenente tutte le CWS configurate nella task selezionata, la loro gerarchia e l'indicazione dei PD dove tali CWS sono istanziate.

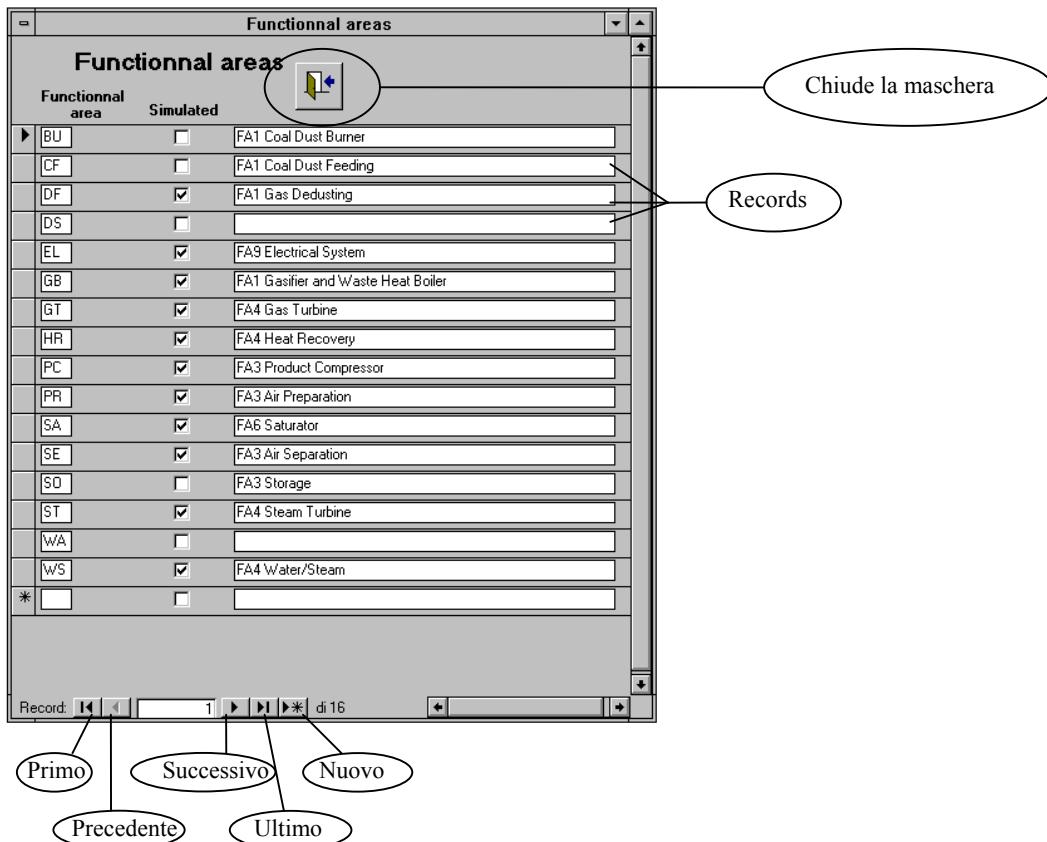
8/ Questi tasti avviano la produzione dei reports di documentazione del progetto, in particolare di:

- specifiche funzionali dell'automazione
- specifiche interfaccia MMI
- test di collaudo di automazione ed MMI

9/ Questo tasto genera un report contenente la lista e la descrizione degli allarmi configurati per ogni task

9.2.1 Tabelle di riferimento

Le tabelle di riferimento contengono dati relativi all'organizzazione generale del progetto come ad esempio la suddivisione in aree, le semplificazioni ammesse, la codifica dei colori da utilizzare nel MMI. Sono caratteristiche del sistema da riprodurre e delle procedure di realizzazione e test stabilite dal responsabile del progetto. Vanno quindi stabiliti generalmente all'inizio del lavoro. Sono accessibili tramite una maschera del tipo di quella della figura seguente.



Developers

Field name	PK	FK	Type	Length	Linked table
Developer name	yes	yes	text	30	RegoPages, ControlWindows, PlantDisplays

Simlevels

Field name	PK	FK	Type	Length	Linked table	Value example
Simulation level	yes	yes	text	2	SiemensPages, PlantDisplays	NS,PS,FS.

Regotasks

Field name	PK	FK	Type	Length	Linked table
Rego task name	yes	yes	text	5	RegoPages, SiemensPages

ProcessTasks

Field name	PK	FK	Type	Length	Linked table
Process task name	yes	yes	text	5	RegoPages

Revisions

Field name	PK	FK	Type	Length	Linked table
Revision	yes	yes	text	2	SiemensPages

FKZs

Field name	PK	FK	Type	Length	Linked table
FKZ	yes	yes	text	6	SiemensPages
Functionnal area	no	yes	text	2	FunctionnalAreas
Simulation level	no	yes	yes/no		

FunctionnalAreas

Field name	PK	FK	Type	Length	Linked table
Functionnal area	yes	yes	text	6	FKZs, PD/FALink
Simulation level	no	yes	yes/no		
Note	no	no	text	50	

Types

Field name	PK	FK	Type	Length	Linked table
Component type	yes	yes	text	4	SiemensPages

SimRulesRef

Field name	PK	FK	Type	Length	Linked table
Simulation rule reference	yes	yes	text	5	SiemensPages
Italian note	no	no	memo		
English note	no	no	memo		

9.2.2 Tabelle dati

Le tabelle dati contengono le informazioni relative al progetto come ad esempio la lista della documentazione utilizzata, la lista delle pagine di regolazione implementate, la configurazione delle CW e dei PD. Per ogni tabella è stata realizzata una maschera che guida all'inserimento dei dati nella forma corretta. Controlli incrociati dell'applicativo impediscono inoltre, per quanto possibile, la duplicazione di dati o l'inserimento di dati errati. Segue un esempio di tabella dati.

Plant Displays / Siemens Pages link					
Siemens Pages / Plant Displays link					
SP KKS code	SP PN	PD KKS name	Simulated	Signal code	
QADA00CG011	1	QADY00EJ001	<input checked="" type="checkbox"/>	XG01	
QADA00CG012	Z1	QADY00EJ001	<input checked="" type="checkbox"/>	XG01	
QADA00CG021	1	QADY00EJ001	<input checked="" type="checkbox"/>	XG01	
QADA00CG022	Z1	QADY00EJ001	<input checked="" type="checkbox"/>	XG01	
QADA01CE001X	01	2MKC00EJ001	<input checked="" type="checkbox"/>	XD01	
QADA01CE005X	01	2MKC00EJ001	<input checked="" type="checkbox"/>	XD01	
QADA01GS011	B02	1MKC00EJ001	<input checked="" type="checkbox"/>	XB01	
QADA01GS011	B02	1MKC00EJ001	<input checked="" type="checkbox"/>	XB02	
QADA01GS012	B02	1MKC00EJ001	<input checked="" type="checkbox"/>	XB01	
QADA01GS012	B02	1MKC00EJ001	<input checked="" type="checkbox"/>	XB02	
QADA02GS011	B02	2MKC00EJ001	<input checked="" type="checkbox"/>	XB01	
QADA02GS011	B02	2MKC00EJ001	<input checked="" type="checkbox"/>	XB02	
QADA02GS012	B02	2MKC00EJ001	<input checked="" type="checkbox"/>	XB01	
QADA02GS012	B02	2MKC00EJ001	<input checked="" type="checkbox"/>	XB02	
OBBA00CE031	Z1	OBBY00EJ001	<input checked="" type="checkbox"/>	XG52	
OBBA00CE032	Z1	OBBY00EJ001	<input checked="" type="checkbox"/>	XG52	
OBBA00CE033	Z1	OBBY00EJ001	<input checked="" type="checkbox"/>	XG52	
OBBA01CG001	Z1	OBBY00EJ001	<input checked="" type="checkbox"/>	XG01	
OBBA01CG002	Z1	OBBY00EJ001	<input checked="" type="checkbox"/>	XG01	
OBBA05CE022	Z1	OBBA00EJ001	<input checked="" type="checkbox"/>	XG01	
OBBA10CE022	Z1	OBBA00EJ001	<input checked="" type="checkbox"/>	XG01	
OBBA10GS001	1	OBBA00EJ001	<input checked="" type="checkbox"/>	XG01	

Segue struttura e sommaria descrizione dei campi contenuti in ogni tabella dati.

NB: PK indica che il campo contiene una chiave primaria (Primary Key in linguaggio ACCESS), mentre i campi FK contengono una chiave esterna (Foreign Key in linguaggio ACCESS), cioè un campo collegato ad un campo di un'altra tabella. Ognuna di queste chiavi definisce delle proprietà o dei vincoli che devono essere soddisfatti dal contenuto del campo stesso.

SiemensPages

Field name	PK	FK	Type	Length	Linked table	Examples
KKS code	yes	yes	text	12	SP/RPlink, ControlWindows, SP/PDlink	0LAB51EG050
Page number	yes	yes	text	4	SP/RPlink, ControlWindows, SP/PDlink	01
Used revision	no	yes	text	2	Revisions	R0
Date of used revision	no	no	date			
Obtainable revision	no	yes	text	2	Revisions	R1
Date of obtainable revision	no	no	date			
Emission date	no	no	date			
Simulation level	no	yes	text	2	SimLevels	NS, PS, FS
Description	no	no	text	50		LPFP1 TEMP F
FKZ	no	yes	text	6	FKZs	
Type	no	yes	text	4	Types	Act, Meas, CLC
Note	no	no	text	50		
Rego task	no	yes	text	8	RegoTasks	OXY, ARIWAS

RegoPages

Field name	PK	FK	Type	Length	Linked table	Value example	Note
Diagram name	yes	yes	text	4	SP/RPLink	L022	
Regotask name	no	yes	text	8	RegoTasks		
Process task	no	yes	text	8	ProcessTasks		
Development date	no	no	date				
Assignment date	no	no	date				
Connecting date	no	no	date				
Developer	no	yes	text	30	Developers	DRAGONI	
Updated	no	no	yes/no				Indica l'avvenuto aggiornamento della RegoPages all'ultima versione disponibile della SiemensPage
Test date	no	no	date				
Tester	no	yes	text	30	Testers	DRAGONI	
Note	no	no	text	50			
Release	no	no	integer	3		1,2,3	

PlantDisplays

Field name	PK	FK	Type	Length	Linked table
KKS	yes	yes	text	12	SP/PDlink, PD/CWlink
Simulation level	no	yes	text	2	SimLevels
Developer	no	yes	text	30	Developers
Development date	no	no	date		
Tester	no	yes	text	30	Testers
Test Date	no	no	date		
Note	no	no	memo		
H1	no	no	integer		
H2	no	no	integer		
H3	no	no	integer		
H4	no	no	integer		
H5	no	no	integer		
H6	no	no	integer		

Simulation rules

Field name	PK	FK	Type	Length	Linked table
Simulation rule reference	yes	yes	text	6	SP/SRRlink
Simulation level	no	no	text	50	
Note (Italian)	no	no	memo		
Note (English)	no	no	memo		

SP/RPLink

Field name	PK	FK	Type	Length	Linked table
KKS code	yes	yes	text	12	SiemensPages
Page number	yes	yes	text	4	SiemensPages
Diagram name	yes	yes	text	4	RegoPages

SP/SRRk

Field name	PK	FK	Type	Length	Linked table
KKS code	yes	yes	text	12	SiemensPages
Page number	yes	yes	text	4	SiemensPages
Simulation rule	yes	yes	text	6	Simulation rule reference

SP/PDlink

Field name	PK	FK	Type	Length	Linked table
SP KKS code	yes	yes	text	12	SiemensPages
Page number	yes	yes	text	4	SiemensPages
PD KKS code	yes	yes	text	12	PlantDisplays
Signal code	yes	no	text	4	
Simulation level	no	no	yes/no		

PD/CWlink

Field name	PK	FK	Type	Length	Linked table
PD KKS code	yes	yes	text	12	PlantDisplays
CW KKS type	yes	yes	text	19	ControlWindows
SCADA link	no	no	text	3	(OK, NOK, NYD)

PD/FALink

Field name	PK	FK	Type	Length	Linked table
PD KKS code	yes	yes	text	12	PlantDisplays
Functional area	yes	yes	text	6	FunctionalAreas

9.2.3 Tabelle Control Window

Il data-base può contenere tutte le informazioni necessarie alla creazione delle CW. La più recente versione dell'applicativo permette sia l'inserimento dei dati tramite un'apposita maschera (che naturalmente sarà diversa a seconda del tipo di stazione da configurare, dal momento che i dati necessari cambiano da CW a CW), sia la generazione dei files che poi verranno elaborati dalle routine LINUX per la creazione delle CW vere e proprie.

La parte superiore della maschera è uguale per tutte le tabelle e contiene i dati descrittivi della CW compresa la gerarchia che, lo ricordiamo, può essere calcolata con l'apposita procedura. La parte inferiore della maschera contiene invece i dati di configurazione tipici della stazione. Alcuni di questi campi possono essere elaborati automaticamente una

volta riempita la prima sezione della maschera anche se tale procedura richiede tempi di esecuzione molto lunghi.

Da questa maschera possono essere richiamate le due ulteriori seguenti maschere:

Utility

Da questa maschera è possibile visualizzare la tabella dati (Open CW), eseguire l'aggiornamento delle CW (cioè riempire automaticamente alcuni dei campi dati una volta inseriti i dati descrittivi, tramite il tasto Update CW), esportare i files necessari alla configurazione delle CW (Exp CW). Le stesse opzioni sono disponibili per la configurazione delle stazioni “Malfunction”



Malfunction



Da questa maschera è possibile configurare tutte le stazioni di MALFUNCTION (che comprendono sia le interfacce di malfunzione gestibili dall'istruttore, sia le stazioni di “REMOTE PARAMETER” per la gestione delle condizioni a contorno)

9.3 Aggiornamento della documentazione sullo stato del simulatore

Il database è in grado di gestire lo stato del simulatore in base alla documentazione disponibile anche durante lo sviluppo del progetto, in funzione delle revisioni del progetto di impianto disponibile per essere implementate nelle task di regolazione o nel MMI

Nella tabella Siemens pages esistono appunto quattro campi destinati a questo scopo:

- “used revision” contiene il codice della revisione (della documentazione di impianto) attualmente implementato nel simulatore
- “date of used revision”
- “obtainable revision” contiene il codice di revisione di documentazione disponibile ma non ancora implementata
- “date of obtainable revision”

Inoltre, nella tabella “Rego pages” i campi “Release” ed “Updated” indicano rispettivamente la versione del simulatore in uso e l’effettivo allineamento del simulatore all’ultima versione di documentazione di impianto disponibile a quel momento.

9.4 Aggiornamento automatico delle tabelle Siemens pages e Rego pages

Qualora nella tabella Siemens Pages venga aggiornato il campo “obtainable revision” con un nuovo valore diverso da quello esistente nel campo “used revision”, il campo “Updated” di tutte le Rego pages collegate verrà automaticamente messo a “no”. Al contrario aggiornando da “no” a “yes” il campo “Updated” della tabella Rego page, il campo “used revision” e “date of used revision” saranno automaticamente sostituiti con i campi “obtainable revision” e “date of obtainable revision” (che resteranno quindi vuote) delle Siemens Pages collegate.

È possibile, ma non consigliabile, eseguire automaticamente l'aggiornamento a “yes” di tutte le Rego pages di una task, attivando quindi la procedura prima descritta, tramite il tasto “UPDATING REGO TASK” presente nella maschera principale.

9.5 Gerarchia di plant displays e control windows

Il concetto di gerarchia è utilizzato dal sistema TELEPERM, ma non solo da questo, per l'organizzazione di due principali funzioni:

- la navigazione nelle pagine video che permette di raggiungere elementi contenuti in pagine video specializzate, partendo da pagine di alto livello e spostandosi rapidamente a quelle di dettaglio senza la necessità da parte dell'operatore di memorizzare il nome o la posizione della pagina cercata
- la separazione in aree degli allarmi e la possibilità di navigare direttamente dalla pagina allarmi (o parimenti dai riassuntivi presenti in ogni pagina) al sinottico dove questo è più efficacemente osservabile dell'operatore. Infatti ogni allarme tramite la gerarchia è assegnato ad una CW

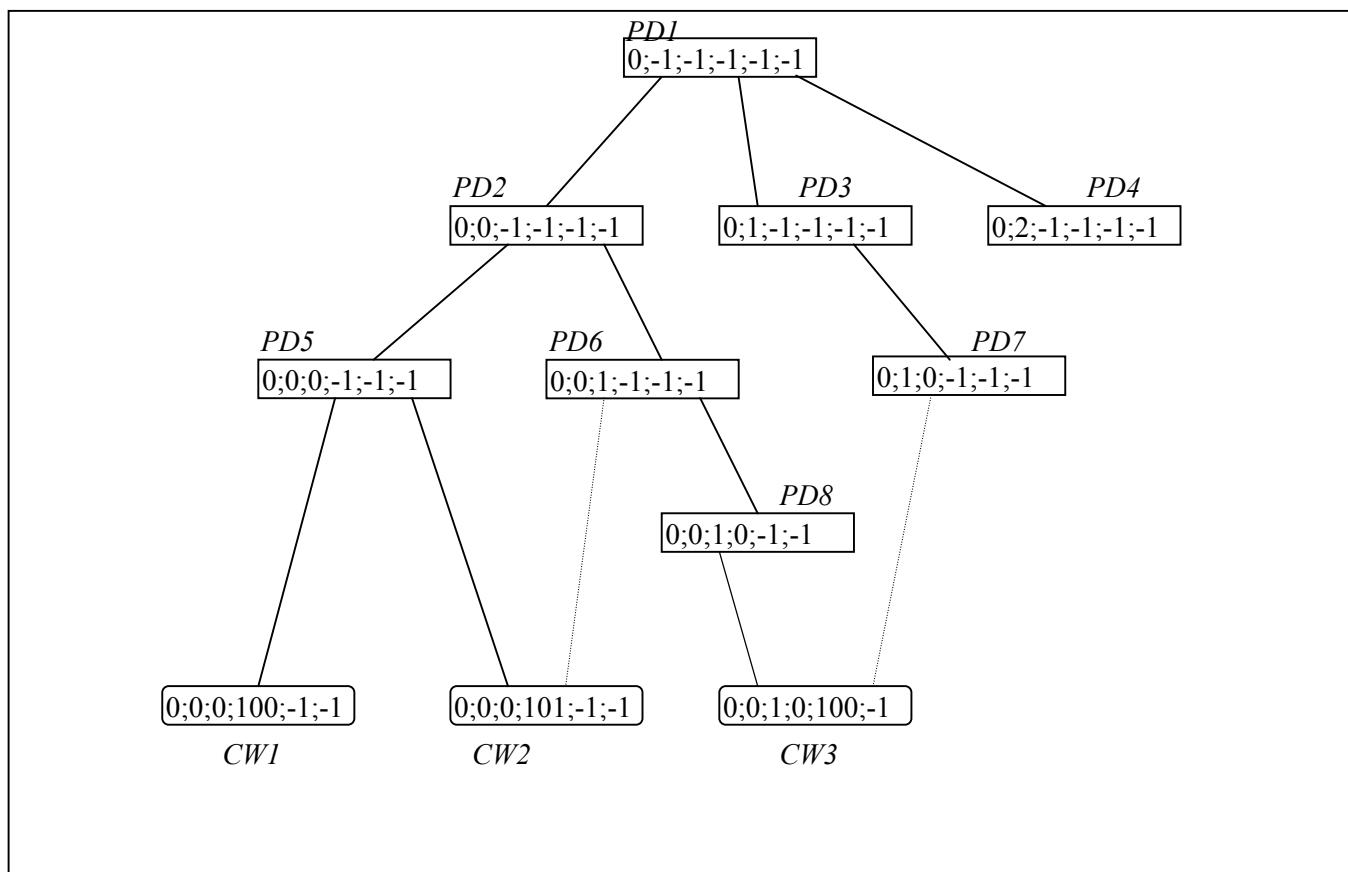
9.5.1 Concetto di gerarchia

La gerarchia è un attributo numerico che deve essere assegnato a tutti gli elementi attivi del MMI, composto da sei cifre.

Il seguente diagramma illustra l'organizzazione della gerarchia tra PD e CW.

Nell'esempio seguente il sinottico PD1 è quello che generalmente contiene i dati salienti di tutto l'impianto. Da qui l'operatore può raggiungere tre ulteriori PD PD2 PD3 e PD4 che, rappresentano solitamente aree impianto. Da ognuno di essi è poi possibile raggiungere ulteriori PD contenenti il dettaglio di sistemi dell'area a cui sono collegati. Chiaramente ogni sinottico può avere un solo “padre” e differenti “figli”.

Le CW si trovano a livello più basso della gerarchia, hanno un solo padre e non possono avere figli. Ciò non significa che una stazione deve risiedere solo in un determinato sinottico, ma che dal punto di vista della navigazione a seguito di allarme, il salto pagina tra allarme, sia partendo del riassuntivo che della pagina allarmi, avviene verso il PD a cui è assegnata la CW associata all'allarme.



9.5.2 Assegnazione della gerarchia

Come già ricordato la gerarchia è un codice numerico di sei cifre, contenute nelle tabelle CW nei campi H1,H2, H3, H4, H5, H6.

La gerarchia delle CW avrà i primi campi uguali a quello del sinottico di cui è figlia, mentre l'ultimo campo significativo (quelli non utilizzati devono avere il valore -1) sarà il suo progressivo. Tale valore, a differenza degli altri, sarà numerato a partire dal numero 100.

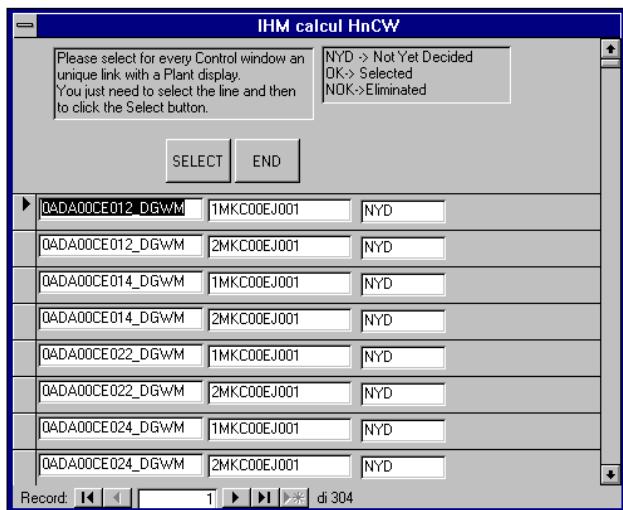
La struttura dell'albero delle gerarchie deve sempre essere completa, vale a dire che deve sempre esistere un sinottico generale (gerarchia 0, -1, -1, -1, -1, -1) e che non può esistere un sinottico di terzo livello (gerarchia 0, 1, 1, -1, -1, -1) se non esiste il corrispettivo padre di secondo livello (gerarchia 0, 1, -1, -1, -1, -1). Naturalmente lo stesso vale per ogni livello di PD.

Mentre la gerarchia dei PD deve essere assegnata manualmente, l'assegnazione delle gerarchie delle CW può essere effettuata tramite una procedura automatica per.

La procedura è attivabile tramite il tasto “Computing the CW hierarchy” della maschera principale. Una volta lanciata la procedura assegna automaticamente la gerarchia alle CW presenti in un unico PD, mentre qualora questo sia presente in diversi PD di pari livello gerarchico, chiede di selezionare a quale di questi va associata la gerarchia della CW in oggetto. Segue un esempio della maschera per l'assegnazione della gerarchia che la procedura automatica presenta nei casi di assegnazioni multiple di una CW.

Nel caso di CW assegnate a PD di diverso livello, verrà automaticamente selezionato quello di livello più basso.

Qualora sia necessario è possibile ricalcolare la gerarchia di una o più CW già assegnata in precedenza mettendo a “no” il campo “Assigned Hyerarchy” nella tabella Control Windows.



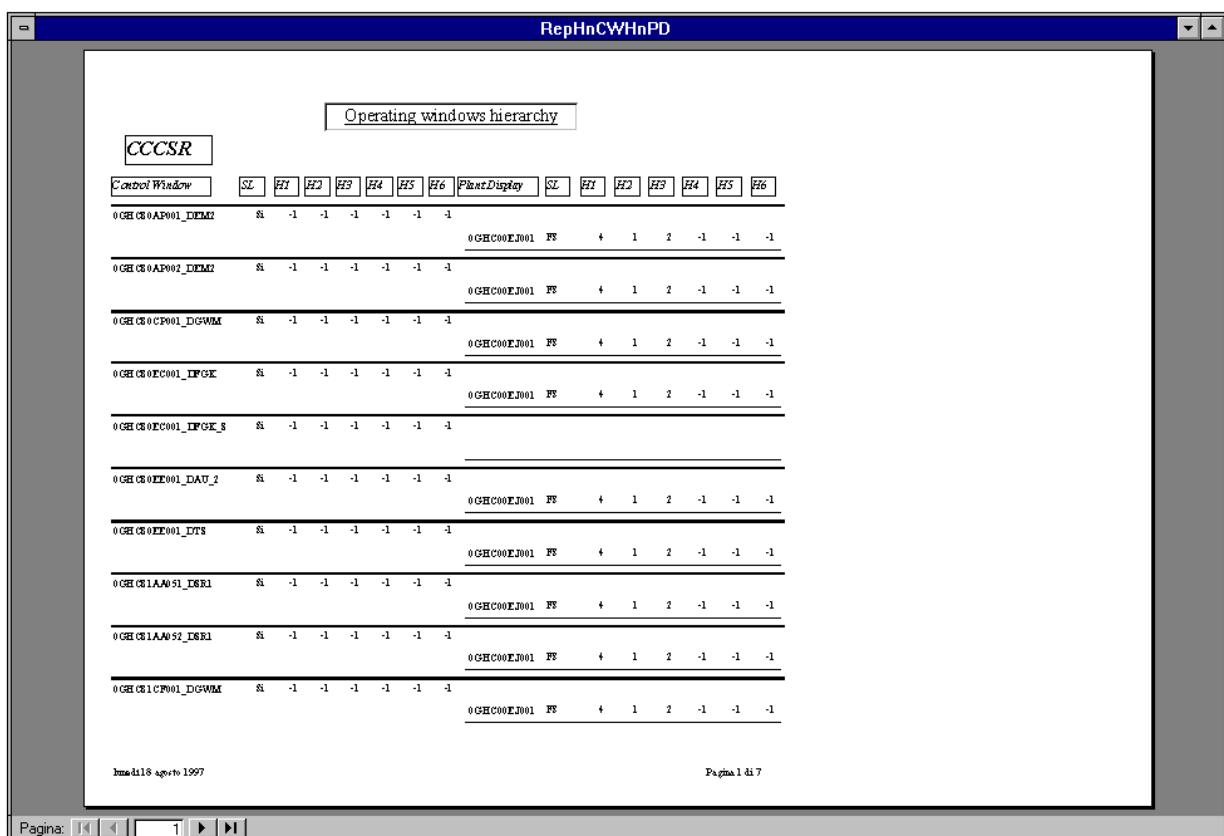
9.6 Documentazione del progetto

Il database permette la creazione automatica di una serie di report che documentano la configurazione del simulatore sia per quanto riguarda l'emulazione del sistema di controllo che per l'MMI ed il sistema SCADA. Il formato di questi report può essere modificato in base alle esigenze del progetto in corso.

9.6.1 Documentazione delle Control Windows

Il report che viene generato, uno per ogni task di regolazione, contiene la lista delle CW con la relativa gerarchia e con i PD a cui sono associate.

Segue un esempio del report prodotto.

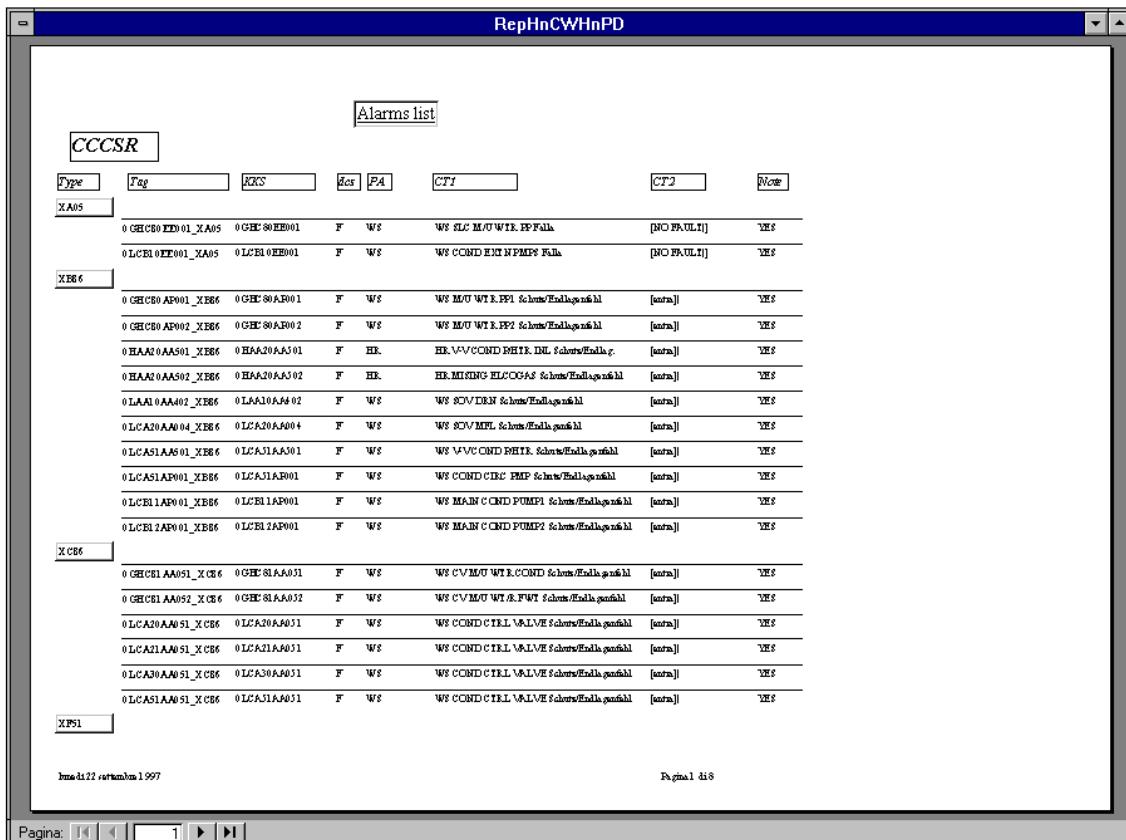


9.6.2 Documentazione degli allarmi

Il report che viene generato, uno per ogni task di regolazione, contiene la lista degli allarmi comprensivi di:

- TAG di allarme
- KKS della pagina di regolazione in cui è presente
- Classe dell'allarme (A, P, T, F)
- Area funzionale a cui è associato
- Campi descrizione dell'allarme

Segue un esempio del report prodotto.



9.6.3 Specifiche funzionali sui diagrammi funzionali

Possono essere generati tre report per ogni area funzionale contenenti principalmente:

- **FKZ** : contiene la lista dei FKZ e l'indicazione del livello di simulazione (FS, PS, NS)
 - **FD** : contiene l'elenco delle Siemens Page e il loro livello di simulazione
 - **FD notes** : contiene le note sulle semplificazioni apportate durante la realizzazione del modello e altre note inserite dal configuratore

9.6.4 Specifiche funzionali sul MMI

Possono essere generati quattro report per ogni area funzionale contenenti principalmente:

- **CW** : lista delle CW e relativo livello di simulazione
 - **PD** : lista dei PD e relativo livello di simulazione
 - **PD/CW** : lista di tutte le CW contenute in ogni PD
 - **PD/SP** : lista dei link diretti tra Siemens pages e PD

9.6.5 Test report

Possono essere generati tre report per ogni Rego task contenenti principalmente:

- **RSTT** : data di esecuzione dei tests sulla singola Rego task
 - **CWT** : data di esecuzione dei tests sulle CW
 - **PDT** : data di esecuzione dei tests sui PD