**SAMA图概要设计文档**

**1 引言**

**1.1 目的**

本文档依据《MACS V65X软件技术改进与开发\_SAMA图需求规格说明书》进行设计，描述了SAMA图实现的关键技术、子系统划分、接口定义。

编写此文档的目的是明确AT软件各功能模块的接口与设计思路，作为详细设计及编码工作的输入，并指导后续的测试和维护工作。

**1.2 预期读者**

本文档的预期读者是：测试人员、开发人员、手册编写人员。

**1.3 术语与缩略语**

AT：AutoThink，HOLLiAS MACS V6 版本中的算法组态软件AutoThink的简称

总控：工程总控

ST：文本语言

**1.4 引用文档**

[1] 《MACS V65X软件技术改进与开发\_SAMA图需求规格说明书》

[2] 《MACS V65X软件技术改进与开发\_SAMA图导出可行性分析报告》

**2 假设和约束**

**2.1假设**

**2.2 约束**

**3 系统概述**

**3.1 系统结构概述**

HOLLiAS MACS V6.5.4系统主要由以下子系统构成：

1. 控制站：包括控制器和I/O卡件，主要实现采集过程数据，进行数据转换和处理，实施连续、批量或顺序控制的运算和输出控制作用。
2. 历史站：包括I/O 服务，实时数据库服务，历史数据库服务，报警处理服务，主要进行历史数据的收集存储、报警、日志记录的产生。
3. 工程师站：包括数据库组态软件、图形组态软件、控制算法组态软件、报表组态软件，主要完成工程、监视画面、控制算法、报表的组态以及一些系统设置。
4. 操作员站：包括操作员在线监视软件，报表打印服务，提供监视、操作现场设备的人机界面。

系统中各个节点通过网络连接，系统划分为：控制网和系统网。



图 1 系统结构图

**4 模块设计**

**4.1 软件结构描述**

AT目前的架构如下图所示：



本次涉及修改的模块如下：

1：框架模块

2：CFC模块

本次新增模块如下：

1：SAMADLL

2：VisioAPI

3：基础工程文件

由于新增模块SAMADLL与VisioAPI，所以AT的架构调整如下图所示：



下文以模块为单位阐述设计方案。

**4.2 框架模块**

**4.2.1 单个控制站级导出**

该功能为单个控制下全部POU或选择部分POU导出SAMA图。

* 导出SAMA图菜单项

在框架的【工程】菜单里面增加“导出SAMA图”菜单项，如下图所示：



对应的菜单资源为：

IDR\_AUTOTHINK\_CFC

IDR\_AUTOTHINK\_LD

IDR\_AUTOTHINK\_SFC

IDR\_AUTOTHINK\_ST

增加的导出SAMA图的菜单项ID为：

IDM\_EXPORT\_SAMA

在框架模块的Resource.h里面增加该ID的定义：

#define IDM\_EXPORT\_SAMA 974

在MainFrm.h定义该消息响应函数：

class CMainFrame : public CMDIFrameWnd

{

……………………………………………

……………………………………………

……………………………………………

protected:

//{{AFX\_MSG(CMainFrame)

……………………………………………

***afx\_msg void OnExportSAMA();***

***afx\_msg void OnUpdateExportSAMA(CCmdUI\* pCmdUI);***

……………………………………………

//}}AFX\_MSG

……………………………………………

};

在MainFrm.cpp定义该消息响应函数：

ON\_COMMAND(***IDM\_EXPORT\_SAMA,OnExportSAMA***)

ON\_UPDATE\_COMMAND\_UI(***IDM\_EXPORT\_SAMA, OnUpdateExportSAMA***) //该函数控制【导出SAMA图】菜单项是否使能。

* 导出SAMA图

函数void OnExportSAMA()的实现流程图如下所示：



* 导出SAMA图对话框

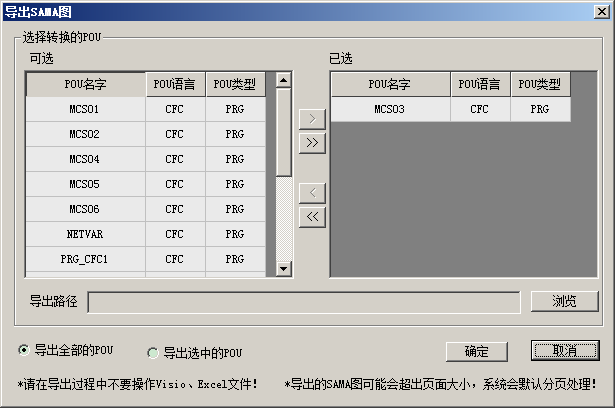
点击导出SAMA图菜单项，弹出导出SAMA图对话框，如下：



**注：导出SAMA图菜单的使能前提条件是该站的工程编译通过。**

POU到Sheet页大小默认与POU页面大小一致

图纸大小不为A3，则提示“导出的SAMA图超出页面大小，系统会默认分页处理”，如下图：



导出的两种方式：

1. 如果不选择转换的POU，直接点击确定，则导出全部的POU，按文件夹层级分级导出，导出路径为工程名文件夹所在的站号下。
2. 如果点击了【导出选中的POU】按钮，则需要选择转换的POU，通过这种方式的操作，所有选择的POU导出为一个VSD文件，每个POU占一个sheet页，导出路径为所选的“导出路径”

增加导出SAMA图的对话框类如下：

class CExportSAMADlg : public CDialog

{

// Construction

public:

CExportSAMADlg(CWnd\* pParent = NULL); // standard constructor

virtual ~CExportSAMADlg();

// Dialog Data

//{{AFX\_DATA(CVISIOTOATXMLDlg)

enum { IDD = IDD\_EXPORT\_SAMA };

//}}AFX\_DATA

// ClassWizard generated virtual function overrides

//{{AFX\_VIRTUAL(CVISIOTOATXMLDlg)

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV support

//}}AFX\_VIRTUAL

// Implementation

protected:

// Generated message map functions

//{{AFX\_MSG(CVISIOTOATXMLDlg)

virtual BOOL OnInitDialog();

virtual void OnOK();

afx\_msg void OnSelectPaper();

afx\_msg void OnBrowse();

afx\_msg void OnRdExpAll();

afx\_msg void OnRdExpCkd();

afx\_msg void OnToRight();

afx\_msg void OnAllToRight();

afx\_msg void OnToLeft();

afx\_msg void OnAllToLeft();

afx\_msg void OnGridEndEdit(NMHDR \*pNotifyStruct, LRESULT\* pResult);//ON\_NOTIFY的消息映射函数

afx\_msg void OnLeftGridSelChanged(NMHDR \*pNotifyStruct, LRESULT\* pResult);//ON\_NOTIFY的消息映射函数

afx\_msg void OnRightGridSelChanged(NMHDR \*pNotifyStruct, LRESULT\* pResult);//ON\_NOTIFY的消息映射函数

//}}AFX\_MSG

DECLARE\_MESSAGE\_MAP()

public:

void GetExportParams(OUT SAMA\_EXPORT\_PARAMS \* pParams);

void SetVSDVision(CString VSDVision);

private:

void UpdateState();

void InitFillGridTable();

void SetExpoutPouGridHeader();

void FillGridTable(CGridCtrl& gridCtrl, set<CString> &strPouArray);

void GetFolderAndPouList(CPOUTreeNode\* pFatherNode, CPouNodeList& tFOLDERNodeList, CPouNodeList& tCFCPOUNodeList); //将根节点pFatherNode下的文件夹和POU放入对应的List中

void GetFolderAndPouList(CPOUTreeNode\* pFatherNode, CPouNodeList& tFOLDERNodeList, std::list<CString /\* POU NAME \*/> & lsPOUNames);

void GetFOLDERPouToSAMA(CPOUTreeNode\* ptPOUTreeNode, std::list<CString /\* POU NAME \*/> & lsPOUNames);

public:

CString m\_SamaSysFilepath;

CString m\_SamaExportFilepath;

bool m\_bExportCheckedPOU;

private:

CStringArray m\_arrSelectPOUs; //用户选择的POU集合

CVApplication m\_pVApplication;

CString m\_strVsdVision;

int m\_icontentindex;

DWORD m\_dwSelCnt;

CFont\* m\_pFont;

CSAMAGridCtrl m\_gridLeft; //grid控件对象

CSAMAGridCtrl m\_gridRight; //grid控件对象

CString m\_strPath; //用户选择的导出路径

set<CString> m\_setLeftPou; // 当前左视图下POU

set<CString> m\_setRightPou; // 当前右视图下POU

CButton m\_btnToRight, m\_btnAllToRight, m\_btnToLeft, m\_btnAllToLeft; // 左右4个按钮

CButton m\_btnRadioAll, m\_btnRadioCkd; // 2个单选框

};

IDD\_EXPORT\_SAMA为该对话框的资源ID号，定义如下：

#define IDD\_EXPORT\_SAMA 1072

该类定义与其成员函数实现分别定义在DlgExportSAMA.h以及DlgExportSAMA.cpp中

导出参数数据结构，定义如下：

// 导出SAMA图类型

enum EN\_EXPORT\_TPYE

{

EN\_ET\_EXPORT\_SELECTED, /\* 导出选择的 \*/

EN\_ET\_EXPORT\_STATION, /\* 导出工作站 \*/

};

// SAMA图导出参数：POU信息

typedef struct SAMA\_DLL\_DECL \_tag\_SAMAExportParam

{

std::list<CString /\* POU NAME \*/> lsPOUNames; /\* POU列表 \*/

CString strSavePath; /\* 文件路径 \*/

void Release() {lsPOUNames.clear();}

}SAMA\_EXPORT\_PARAM, \*PSAMA\_EXPORT\_PARAM;

typedef std::list<PSAMA\_EXPORT\_PARAM> PSAMA\_EXPORT\_PARAM\_LIST; /\* 导出参数列表类型定义 \*/

// SAMA图导出参数

typedef struct SAMA\_DLL\_DECL \_tag\_SAMAExportParams

{

EN\_EXPORT\_TPYE enType; /\* 导出类型 \*/

unsigned int uiPouCount; /\* 导出的POU个数 \*/

CString strSaveFolder; /\* 导出目录 \*/

CString strFileEx; /\* 导出文件后缀名 \*/

CString strFileName; /\* 导出文件名 \*/

CString strSAMAFilePath; /\* SAMA图文件路径 \*/

PSAMA\_EXPORT\_PARAM\_LIST lsExportParams; /\* 参数列表 \*/

ISAMACallBack \* pSamaCallBack; /\* SAMA图操作回调对象 \*/

void Release();

}SAMA\_EXPORT\_PARAMS, \*PSAMA\_EXPORT\_PARAMS;

* SAMA图导出界面初始化流程

函数BOOL CExportSAMADlg::OnInitDialog()的实现流程图如下：



* SAMA图导出界面确定流程

函数void CExportSAMADlg::OnOK()的实现流程图如下：



* POU转换为SAMA图

函数void CSAMA::TransPouToSama(const SAMA\_EXPORT\_PARAMS \* pParams) 的实现流程图如下：



* 导出EXCEL交叉引用信息

SAMA图EXCEL导出的交叉引用表，本质上就是超链接的信息导出。因为SAMA图上图元的交叉引用信息颗粒度到POU级别，并且只针对输入图元和输出图元，但是超链接针对的是输入输出以及块元件的引用信息。静态的索引表要求每个变量在POU里面的引用情况。所以针对该需求，需要导出的是SAMA图元的超链接信息。

函数void CSAMA::ExcelPouCrossReference ()处理流程图如下：



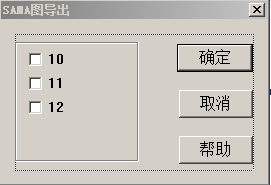
* 导出EXCEL功能块参数表信息

函数void CSAMA::ExcelPouparameter ()处理流程图如下：



**4.2.2 总控导出SAMA图**

在总控上，也可以针对本域进行所选的站进行导出。以工程和控制站为单位导出时，入口在工程总控处，点击菜单工程/导出SAMA图，弹出如下对话框：



用户选择对应的控制站后，总控记录对应的站号信息，然后在总控编译阶段，由总控依次启动每个站，如果该站已经选择为导出SAMA图，则总控给AT发送一个导出SAMA图的交互ID，AT在对应的消息ID里面导出该站的SAMA图。

在HMIInterface.h里面定义与总控交互的导出SAMA图消息，定义如下：

#define ID\_EXPORT\_SAMA 234

void OnMessageExportSama();

该函数的处理流程图如下：



其中导出SAMA图对话框的ONOK函数处理流程见4.2.1节。

其中【导出SAMA图】对话框以及【选择POU】对话框资源均定义在AutoThink.exe中。

**4.3 CFC模块**

由于CFC模块只是针对CFC语言的一些功能，为了不影响现有CFC模块的稳定性，为了实现导出SAMA图功能与现有的CFC模块解耦，只需要对CFC语言的一些类进行导出设置即可。所有的导出功能均在SAMADLL模块里面实现。

CFC需要设置导出的类如下：

/////////////////////////////////////////////////////////////////////////////////导出类宏定义

#undef CFC\_DLL\_DECL

#ifdef \_CFC\_DLL\_DECL

#define CFC\_DLL\_DECL \_declspec(dllexport)

#else

#define CFC\_DLL\_DECL \_declspec(dllimport)

#endif

/////////////////////////////////////////////////////////////////////////////////导出的元件类

class CFC\_DLL\_DECL CCFCElement : public CElement

class CFC\_DLL\_DECL CCFCBox : public CCFCElement

class CFC\_DLL\_DECL CCFCInput : public CCFCElement

class CFC\_DLL\_DECL CCFCOutput : public CCFCElement

class CFC\_DLL\_DECL CCFCLabel : public CCFCElement

class CFC\_DLL\_DECL CCFCReturn : public CCFCElement

class CFC\_DLL\_DECL CCFCJump : public CCFCElement

class CFC\_DLL\_DECL CCFCComment : public CCFCElement

/////////////////////////////////////////////////////////////////////////////////导出的引脚类

class CFC\_DLL\_DECL CCFCPin : public CObject

class CFC\_DLL\_DECL CCFCOutputPin : public CCFCPin

class CFC\_DLL\_DECL CCFCInputPin : public CCFCPin

//导出连线类

class CFC\_DLL\_DECL CCFCLine : public CObject

**4.4 SAMADLL模块**

SAMADLL为新增模块，所有的SAMA图翻译均在该模块里面实现，其类图如图4.X



图4.X SAMADLL模块类图

类可以分为三大类，接口类（或数据结构）、流程控制类、元件转换类。

**4.4.1 接口类设计**

接口类是对外提供数据交换的类接口，定义在../ 01\_Include/08.IEC/SAMA/SAMA.h中，包含交换数据结构的定义和类接口的定义，接口类在类图的标注如图4.X



图4.X 接口类标注类图

**导出类型枚举型定义**

enum EN\_EXPORT\_TPYE

{

EN\_ET\_EXPORT\_SELECTED, /\* 导出选择的 \*/

EN\_ET\_EXPORT\_STATION, /\* 导出工作站 \*/

};

**导出参数的数据结构定义**

typedef struct SAMA\_DLL\_DECL \_tag\_SAMAExportParam

{

std::list<CString /\* POU NAME \*/> lsPOUNames; /\* POU列表 \*/

CString strSavePath; /\* 文件路径 \*/

void Release() {lsPOUNames.clear();} /\* 对外提供数据清理函数 \*/

}SAMA\_EXPORT\_PARAM, \*PSAMA\_EXPORT\_PARAM;

**导出参数列表的类型定义**

typedef std::list<PSAMA\_EXPORT\_PARAM> PSAMA\_EXPORT\_PARAM\_LIST;

**导出参数**

typedef struct SAMA\_DLL\_DECL \_tag\_SAMAExportParams

{

EN\_EXPORT\_TPYE enType; /\* 导出类型 \*/

unsigned int uiPouCount; /\* 导出的POU个数 \*/

CString strSaveFolder; /\* 导出目录 \*/

CString strFileEx; /\* 导出文件后缀名 \*/

CString strFileName; /\* 导出文件名 \*/

CString strSAMAFilePath; /\* SAMA图文件路径 \*/

PSAMA\_EXPORT\_PARAM\_LIST lsExportParams; /\* 参数列表 \*/

ISAMACallBack \* pSamaCallBack; /\* SAMA图操作回调对象 \*/

void Release(); /\* 内存释放 \*/

}SAMA\_EXPORT\_PARAMS, \*PSAMA\_EXPORT\_PARAMS;

**SAMA图操作回调方法抽象**

class SAMA\_DLL\_DECL ISAMACallBack

{

public:

virtual void SetProcessPos(unsigned int Pos) = 0; /\* 设置处理进度 \*/

};

**SAMA图操作接口抽象**

class SAMA\_DLL\_DECL ISAMA

{

public:

/\* 设置Visio Application \*/

virtual void SetCVApplication() = 0;

/\* 设置系统路径 \*/

virtual void SetSamaSysFilepath() = 0;

/\* 设置导出的参数 \*/

virtual void SetExportParams(SAMA\_EXPORT\_PARAMS \* pParam) = 0;

/\* 开始导出 \*/

virtual void StartTransSama() = 0;

/\* 获取SAMA图操作的单件对象 \*/

static ISAMA \* GetSAMASingleton();

/\* 导出交叉引用 \*/

virtual void ExcelPouCrossReference() = 0;

/\* 导出功能块引用表 \*/

virtual void ExcelPouparameter() = 0;

/\* 设置SAMA图导出占比 \*/

virtual void SetSamaRange(int val) = 0;

};

**4.4.2 流程控制类设计**

流程控制类包含CSAMA和CSAMAPOU两个类，其中CSAMA函数继承自ISAMA接口类，主要功能职责包括、实现接口函数、解释导出参数、控制POU导出逻辑和目录生成；CSAMAPOU类是具体的POU导出控制类，所有的POU导出需要通过CAMAPOU实现，类定义在文件夹../05\_IEC/07.SAMA/SAMASource下，流程控制类在类图的标注如图4.X



图4.X 流程控制类标注类图

CSAMA类元件转换控制流程如图4.X：



图4.X CSAMA类元件转换控制流程图

CSAMAPOU类转换流程如图4.X和4.X：



图4.X POU元件转换流程图1



图4.X POU元件转换流程图2

**4.4.1 元件转换类设计**

元件转换类主要负责CFC元件到SAMA图元件之间的转换，其中，CSAMAElement是元件基类，提供基本的元件转换方法，根据不同的CFC元件类型，七大类元件分为七个类继承自CSAMAElement类元件，转换类在类图中的标注如图4.X所示：

图4.X元件转换类标注类图

类定义在文件夹../05\_IEC/07.SAMA/SAMASource下，说明注释参见表4.X。

|  |  |  |
| --- | --- | --- |
| 类名 | 说明 | 文件名 |
| CSAMAElement | 元件基类，元件转换的基本方法，转换流程参照图4.X的构建图元子流程。 | SAMAElement.cpp  SAMAElement.h |
| CSAMABox | 块元件转换成SAMA图元，其中包含特殊块的构建。 | SAMABox.cpp  SAMABox.h |
| CSAMAComment | 注释元件转换 | SAMAComment.cpp  SAMAComment.h |
| CSAMAInput | 输入元件转换类 | SAMAInput.cpp  SAMAInput.h |
| CSAMAJump | 跳转元件转换类 | SAMAJump.cpp  SAMAJump.h |
| CSAMALabel | label元件转换 | SAMALabel.cpp |
| CSAMAReturn | Return元件转换 | SAMALabel.h |
| CSAMAOutput | 输出元件转换 | SAMAOutput.cpp  SAMAOutput.h |

表4.X 元件转换类功能说明表

**4.5 VisioAPI模块**

VisioAPI模块是对Visio软件提供的COM接口进行了一个扩展MFC动态链接库封装。以C++类的组织方式进行了常用接口的封装，并对这些类进行了导出。

其导出宏如下：

#undef VISIOAPI\_DLL\_DECL

#ifdef \_VISIOAPI\_DLL\_DECL

#define VISIOAPI\_DLL\_DECL \_declspec(dllexport)

#else

#define VISIOAPI\_DLL\_DECL \_declspec(dllimport)

#endif

Visio COM 接口的C++类封装如下：

class VISIOAPI\_DLL\_DECL CVApplication : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVShapes : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVShape : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVWindow : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVDocuments : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVDocument : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVPages : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVPage : public COleDispatchDriver

class VISIOAPI\_DLL\_DECL CVMasters : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVMaster : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVCell : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVSelection : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVRow : public ColeDispatchDriver

class VISIOAPI\_DLL\_DECL CVSection : public ColeDispatchDriver

…………………………………………………………………………//待定封装类

…………………………………………………………………………//待定封装类

…………………………………………………………………………//待定封装类

**4.6 基础工程文件**

基础工程目录为..\HOLLiAS\_MACS\ENG\BaseProj\BASEPROJ\_Thermal Power\SAMAFiles。

SAMAFiles文件夹下，包含三个子文件夹：INI、VSS、Module。其中，INI文件夹里面存储SAMA图的一些配置文件，VSS文件夹里面存储SAMA图元的模具文件，Module目录下保存模板文件。如图4.X所示：

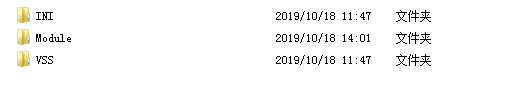


图4.x SAMAFiles目录文件结构图

基础工程文件分为三个部分，SAMA图配置文件、SAMA图模板文件、SAMA模具文件。

**4.6.1 SAMA配置文件**

SAMA配置文件为SAMA图元参数表配置文件，该配置文件的名称为FBPara.ini，保存在基础工程目录下的INI目录下。配置文件针对的是功能块变量，该配置文件配置了类型，对应的参数项以及参数项的初始值。以PID为例，格式如下：

[HSPID]

Count =7

NameValue0=OB|100.000000

NameValue1=OT|100.000000

NameValue2=PT|100.000000

NameValue3=TI|30.000000

NameValue4=TD|0.000000

NameValue5=KD|1.000000

NameValue6=IC|0.000000

**4.6.2 SAMA模板文件**

Module中保存SAMA.vst模板文件，模板文件中包含A0、A2、A3、AX等4个默认页，默认页定义了施工图模板，程序根据POU的纸张大小自动选择使用对应默认页添加页保存SAMA图逻辑。

**4.6.3 SAMA模具文件**

SAMA模具文件的制作，根据需求目前需要做四个模具文件：

* 基础模具文件，是包含普通操作符块、函数块以及功能块模具的文件，如AND、OR等。该模具文件命名为BaseBlock.VSS。
* 特殊功能块模具文件，如PID，MAN等，该模具文件命名为SpecialBlock.VSS。
* 输入输出模具文件，包含输入输出元件对应的模具文件，该模具文件命名为INOUT.VSS。
* 积木模具文件，是构造普通操作符块、函数块以及普通功能块覆盖不到的一些块，如6个输入管脚的AND、6个输入管脚的OR等元件，其中该模具文件命名为BuildingBlock.VSS。