# SFC数据结构设计

## SFC POU 类图



图1 SFC POU类图

## SFC元件类结构图



图2 SFC元件类结构图

## SFC逻辑存储类图



图3 SFC逻辑存储类图

## SFC关联的动作\转换类图



图4 SFC关联的动作\转换类图

# SFC数据结构交互关系

## 文档视图结构与SFC的关联



图5 文档视图结构与SFC的关联

## SFC POU内部的关联



图6 SFC POU内部的关联

# 相关操作

## 选中

选中操作是SFC进行编辑操作的基础。选中后，再能进行插入、复制、剪切、粘贴等编辑操作。

### 选中操作

1. 单选
2. 多选

### 选中实现

鼠标左键响应消息，记录鼠标选中的左上角、右下角位置，获取选中范围内逻辑块，填充如下结构：

*class CSFCMultiSelected : public CObject*

*{*

*public:*

*SFC\_SELECT\_TYPE m\_eSelectedBegin; //选中逻辑块链表开始类型*

*SFC\_SELECT\_TYPE m\_eSelectedEnd; //选中逻辑块链表结束类型*

*LPSFCLogicBlockList m\_pSelectedLogicBlockList;//选中逻辑块链表*

*BOOL m\_bSelectedFlow; //是否选中流*

*CSFCFlow \* m\_pflow; //块链表所属的流或被选中的整条流分支*

*CSFCLogicBlock \* m\_pLogicBlock; //被选中的整条分支所属的块*

*};*

参数意义说明：

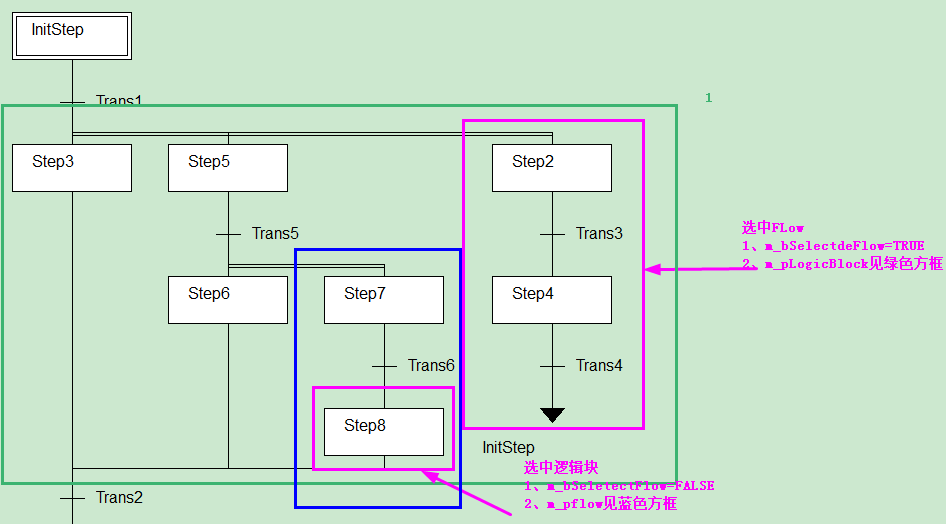


图7 选中结构说明

### 选中类型判断

根据填充好的*CSFCMultiSelected*，解析选中的类型。

有两种情况：

1. 选中单个逻辑块

类型分类如下：

enum SFC\_SELECT\_TYPE

{

ST\_BLOCK\_SIMPLE\_STEP, //简单逻辑块的普通步

ST\_BLOCK\_SIMPLE\_INITSTEP, //简单逻辑块的初始步

ST\_BLOCK\_SIMPLE\_TRANSITION, //简单逻辑块的转换

ST\_BLOCK\_SIMPLE\_JUMP, //简单逻辑块的跳转

ST\_BLOCK\_PARALLEL, //并行逻辑块

ST\_BLOCK\_ALTERNATIVE, //选择逻辑块

ST\_BLOCK\_ERROR //其它类型

};

1. 选中多个逻辑块

需要根据多个逻辑块的首、尾逻辑块类型，判断出选中类型。

//选中逻辑块链表的类型

enum SFC\_BLOCKLIST\_TYPE

{

STEP\_STEP,

TRANSITION\_TRANSITION,

STEP\_TRANSITION,

TRANSITION\_STEP,

BLOCKLIST\_ERROR,

JUMP\_JUMP,

STEP\_JUMP,

TRANSITION\_JUMP

};

表1 多选类型

|  |  |  |
| --- | --- | --- |
| 序号 | 类型 | 含义 |
| 1 | STEP\_STEP | 以步开始，以步结束  C:\Users\FUTING~2\AppData\Local\Temp\063f9391-4e34-4c5e-912b-eb54d8b3b727.tmp |
| 以步开始，以并行分支结束  C:\Users\FUTING~2\AppData\Local\Temp\3674d2da-3481-44c0-9a15-8772dfd8694c.tmp |
| 以步开始，只有步这一简单逻辑块  C:\Users\FUTING~2\AppData\Local\Temp\7795b262-b0b0-4158-9770-89c79d5000cc.tmp |
| 以并行分支开始，以步结束  C:\Users\FUTING~2\AppData\Local\Temp\cc926e72-233a-4ffd-8a0b-9da757025fc2.tmp |
| 以并行分支开始，以并行分支结束  C:\Users\FUTING~2\AppData\Local\Temp\8b3fded5-52b0-43c0-85bf-b55bce9adc92.tmp |
| 以并行分支开始，并只有PARALLEL这一类型逻辑块  C:\Users\FUTING~2\AppData\Local\Temp\9369bf4d-dac9-40ed-9780-b377e57cf854.tmp |
| 2 | TRANSITION\_TRANSITION | 选中单个转换  C:\Users\FUTING~2\AppData\Local\Temp\8a5a1896-e3c3-449f-8fa0-64df910767a2.tmp |
| 以转换开始，以转换结束  C:\Users\FUTING~2\AppData\Local\Temp\b2c05a59-14ff-4b5b-84e2-20375daa5f28.tmp |
| 以转换开始，以选择分支结束  C:\Users\FUTING~2\AppData\Local\Temp\516d6292-a9bd-4914-9d08-6bbf7b1eae59.tmp |
| 以选择分支开始，以转换结束  C:\Users\FUTING~2\AppData\Local\Temp\2a332bf0-28b3-472e-85e0-986f89e0eb1b.tmp |
| 以选择分支开始，以选择分支结束  C:\Users\FUTING~2\AppData\Local\Temp\e683c62e-c6ac-4f82-a7ca-2c3b10b4c139.tmp |
| 以选择分支开始并只有ALTERNATIVE这一类型逻辑块  C:\Users\FUTING~2\AppData\Local\Temp\56e4b14a-5446-40f0-b72f-63eae2dc87e1.tmp |
| 3 | STEP\_TRANSITION | 以步开始，以转换结束  C:\Users\FUTING~2\AppData\Local\Temp\c91e20cd-567c-4d77-aa05-d69a8f5c674b.tmp |
| 以步开始，以选择分支结束  C:\Users\FUTING~2\AppData\Local\Temp\0d40fd44-5690-4580-9061-ed5305dae2f3.tmp |
| 4 | TRANSITION\_STEP | 以转换开始，以步结束  C:\Users\FUTING~2\AppData\Local\Temp\17986727-396f-4018-84ec-cb8e146b8814.tmp |
| 以选择分支开始，以步结束  C:\Users\FUTING~2\AppData\Local\Temp\111f865d-0ed4-47f4-b881-da5b21277529.tmp |
| 5 | BLOCKLIST\_ERROR | 错误处理 |
| 6 | JUMP\_JUMP | 并行分支中有一个分支是跳转 |
| 选中单个跳转  C:\Users\FUTING~2\AppData\Local\Temp\1b818674-30e2-45a9-a6ce-e41f2f5470d8.tmp |
| 7 | STEP\_JUMP | 以步开始，以跳转结束  C:\Users\FUTING~2\AppData\Local\Temp\6f49c907-11a6-45fa-a98a-9346f39fe3d2.tmp |
| 以步开始，以并行分支结束且并行分支均为跳转元件结束  C:\Users\FUTING~2\AppData\Local\Temp\8b219a01-afd8-454b-9251-2b66b00941bc.tmp |
| 以并行分支开始，以跳转结束  C:\Users\FUTING~2\AppData\Local\Temp\fa32d599-810f-435f-b764-69f7f8241abc.tmp |
| 以并行分支开始，以并行分支结束且并行分支均为跳转元件结束  C:\Users\FUTING~2\AppData\Local\Temp\8685dafe-85ee-408f-aaba-63574652a00f.tmp |
| 8 | TRANSITION\_JUMP | 以转换开始，以跳转结束  C:\Users\FUTING~2\AppData\Local\Temp\faa05bae-e4c3-4ae1-a1f8-ba7ef1aaeae1.tmp |
| 以选择分支开始，以跳转结束  C:\Users\FUTING~2\AppData\Local\Temp\3943f09b-4aff-473f-a442-d009ec47e61b.tmp |
| 以转换开始，以并行分支结束且并行分支均为跳转元件结束  C:\Users\FUTING~2\AppData\Local\Temp\618b76da-2e49-4d03-b21a-16364a7e134f.tmp |
| 以选择分支开始，以并行分支结束且并行分支均为跳转元件结束  C:\Users\FUTING~2\AppData\Local\Temp\4373adb8-608c-4d24-a623-c144fb927e55.tmp |

## 插入相关

重点：1、选中元件类型判断

2、菜单使能处理

3、插入操作及视图刷新

**插入操作**—1）步-转换（前）

2）步-转换（后）

3）并行分支（左）

4）并行分支（右）

5）选择分支（左）

6）选择分支（右）

7）跳转

8）转换-跳转

**插入相关菜单使能**

表二 菜单使能控制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 步-转换（前） | 步-转换（后） | 并行分支（左） | 并行分支（右） | 选择分支（左） | 选择分支（右） | 跳转 | 转换-跳转 |
| STEP\_STEP | √ | √ | √ | √ |  |  |  | √ |
| TRANSITION\_TRANSITION | √ | √ |  |  | √ | √ | √ | √ |
| STEP\_TRANSITION | √ | √ |  |  |  |  |  |  |
| TRANSITION\_STEP | √ | √ |  |  |  |  |  |  |
| JUMP\_JUMP | √ |  |  |  |  |  |  |  |
| STEP\_JUMP | √ |  |  |  |  |  |  |  |
| TRANSITION\_JUMP | √ |  |  |  |  |  |  |  |

*其中，红色标注的可能会有特殊处理*

**插入相关命令类：**

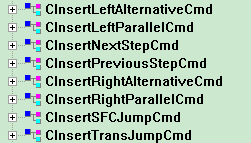




图8 命令类派生关系

**插入基本处理流程**

1. 创建插入命令类
2. 保存信息，创建数据结构
3. 执行插入命令
4. 调整撤销恢复链表
5. 更新POU显示
6. 更新选中区域
7. 设置POU改变相关标志

## 动作、转换相关

相关操作---1）添加\移除入口动作

2）添加\移除出口动作

3）添加\移除步动作

4）添加\移除转换

5）工程管理树，新建\删除\重命名动作POU

管理—CPOUContainer

class CPOUContainer : public CObject

{

DECLARE\_SERIAL(CPOUContainer)

public:

//POU名称数组

CStringArray m\_POUNameArray;

//POU名称和POU对象的MAP

CMap<CString,LPCTSTR,CPOU\*,CPOU\*> m\_POUMap;

};

class CSFCPOU : public CIECPOU

{

public:

CPOUContainer m\_SFCPOUContainer;

};

底层--CPOU

class CPOU : public CObject

{

private:

CPOU\* m\_pFatherPOU;

CMap<CString,LPCTSTR,CPOU\*,CPOU\*>\* m\_pChildPOUMap;

};

1. 命名规则：
2. IEC动作：挂在工程管理树上的

SFCPOU名.用户命名

1. 步动作：双击步添加

SFCPOU名.AT\_元件ID

1. 入口动作

SFCPOU名.AT\_元件ID\_Entry

1. 出口动作

SFCPOU名.AT\_元件ID\_Exit

1. 转换POU

SFCPOU名.AT\_元件ID

问题：打开POU视图，文档标题不符合用户习惯，用户体验不好。

解决：FA\_AT增加变量显示名称：

1. IEC动作：挂在工程管理树上的

SFCPOU名.用户命名

1. 步动作：双击步添加

SFCPOU名.步名

1. 入口动作

SFCPOU名.步名\_Entry

1. 出口动作

SFCPOU名. 步名\_Exit

1. 转换POU

SFCPOU名.转换元件文本

引发问题：

1. 重名情况？关闭、激活文档视图，不能仅通过标题来判断了，还需要POU指针。
2. 特殊字符
3. 字符长度限制？
4. 序列化
5. 存储位置：SFC POU的pou文件中
6. 文件读取：反序列化后，要设置CPOU:: m\_pFatherPOU

*//9月2日*

## 复制、剪切、粘贴

1. 剪贴板格式：SFC\_Clipboard
2. 复制：剪贴板序列化
3. 选中范围中步、转换元件个数
4. 当前POU名称（*why？*）
5. 选中类型及选中块个数
6. 选中范围中步动作、入口动作、出口动作、转换POU搜集及序列化
7. 选中范围中SFC逻辑序列化
8. 剪切

等同于删除

1. 粘贴
2. 判断元件、动作转换POU是否超限？
3. 根据剪贴板中存储的类型和当前选中类型判断是粘贴在后面还是粘贴到右面

4.1 粘贴在后面

4.2 粘贴在右边

*//9月9日*

# 语法检查及翻译

# 差异对比

1. 步

|  |  |  |  |
| --- | --- | --- | --- |
|  | IEC标准 | AT | Codesys V2.3 |
| Step.x | √ | √ | √ |
| Step.\_x | - | √ | √ |
| Step.t | √ | √ | √ |
| Step.\_t | - | √ | √ |
| Step.x/Step.t被赋值 | 应是出错 | 动作\转换POU支持，其它不支持 | √ |
| 最大时间 | - | - | - |
| 最小时间 | - | √ | √ |

1. 转换

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | IEC标准 | AT | Codesys V2.3 | Codesys V3.5 |
| 组态语言 | IL、FBD、ST、LD | LD、ST、CFC | LD、ST、IL、FBD | LD、ST、IL、FBD、CFC、基于页面的顺序功能块图 |
| 组态形式--POU | 单BOOL表达式 | 若干语句;  TRANS:=单BOOL表达式; | 单BOOL表达式 | 转换关联变量:=单BOOL表达式;  （国标规定形式） |

1. 跳转

IEC标准无此元件。AT参照Codeysys。

1. 动作

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | IEC标准 | AT | Codesys V2.3 | Codesys V3.5 |
| 组态语言 | | IL、FBD、ST、LD | LD、ST、CFC | LD、ST、IL、FBD、CFC | LD、ST、IL、FBD、CFC、基于页面的顺序功能块图 |
| IEC步 | 限定符 | 12种  N、S、R、P、P0、P1、SD、DS、SL、L、D、无限定符 | 11种  N、S、R、P、P0、P1、SD、DS、SL、L、D | 9种  N、S、R、P、SD、DS、SL、L、D | 9种  N、S、R、P、SD、DS、SL、L、D |
| 实现形式 | 1. 带最后扫描 2. 不带最后扫描 | 不带最后扫描 | 带最后扫描 | 未知 |
| 动作形式 | 1. 动作POU 2. BOOL变量 | 动作POU | 1. 动作POU 2. BOOL变量 | 1. 动作POU 2. BOOL变量 |
| 普通步 | | - | √ | √ | √ |
| 入口动作 | | - | √ | √ | √ |
| 出口动作 | | - | √ | √ | √ |

1. SFC隐含变量标志

国标中无此描述，Codeys实现更广，包括：最大时间、错误等

|  |  |  |  |
| --- | --- | --- | --- |
|  | IEC标准 | AT | Codesys V2.3 |
| SFCinit | - | √ | √ |
| SFCReset | - | √ | √ |
| SFCPause | - | √ | √ |

1. 不安全、不可到达的SFC组态

IEC标准规定要把这类组态视为出错，AT、codesys2.3均未处理。