

## Express Dictionary Classes

Dictionary of Express information

### Registry

HashTable primordialSwamp; // Dictionary of entities i.e. EntityDescriptors  
 HashTable active\_schemas; // Dictionary of schemas i.e. SchemaDescriptors  
 HashTable active\_types; // Dictionary of types i.e. TypeDescriptors

### SchemaDescriptor

const char \* \_name;

Dictionary entry for a Schema

### TypeDescriptor

const char \* \_name;  
 BASE\_TYPE \_fundamentalType;  
 const TypeDescriptor \* \_referentType;  
 const char \* \_description;

Dictionary entry for a Type.

### AttrDescriptor

const char \* \_name;  
 const TypeDescriptor \* \_domainType;  
 SdaiLogical \_optional;  
 SdaiLogical \_derived;  
 const EntityDescriptor & \_owner;

Dictionary entry for an Attribute.

Dictionary entry for an Entity.  
 EntityDescriptor is derived from TypeDescriptor.

### TypeDescriptor

const char \* \_name;  
 BASE\_TYPE \_fundamentalType;  
 const TypeDescriptor \* \_referentType;  
 const char \* \_description;

### EntityDescriptor

const SchemaDescriptor \* \_originatingSchema;  
 SdaiLogical \_abstractEntity;  
 EntityDescriptorList \_subtypes;  
 EntityDescriptorList \_supertypes;  
 AttrDescriptorList \_explicitAttr;

List of EntityDescriptors and  
 List of AttributeDescriptors.  
 Implemented as a linked list of nodes.

### EntityDescriptorList

#### EntityDescLinkNode

EntityDescriptor \*

#### EntityDescLinkNode

EntityDescriptor \*

#### EntityDescLinkNode

EntityDescriptor \*

### AttrDescriptorList

#### AttrDescLinkNode

AttrDescriptor \*

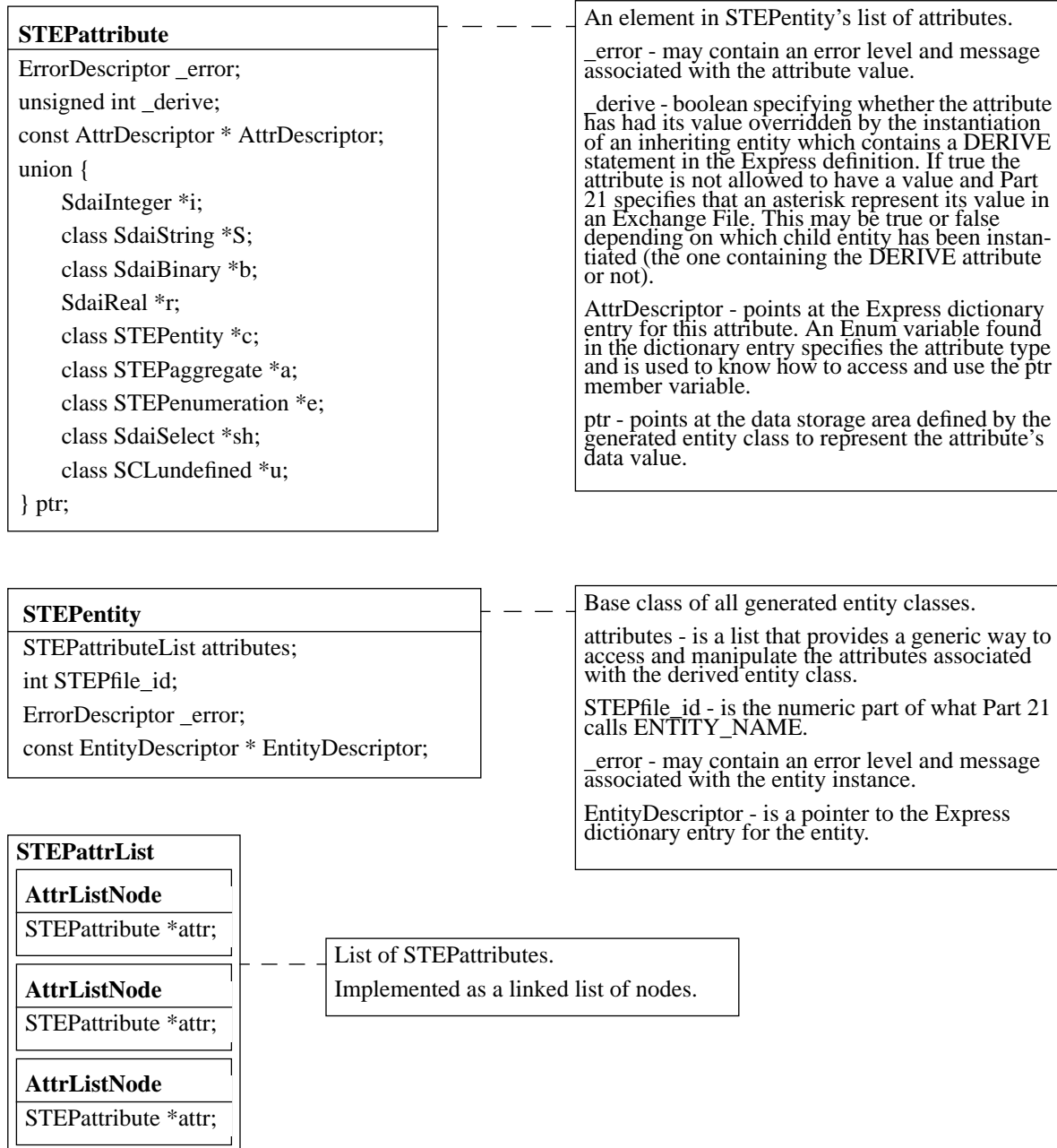
#### AttrDescLinkNode

AttrDescriptor \*

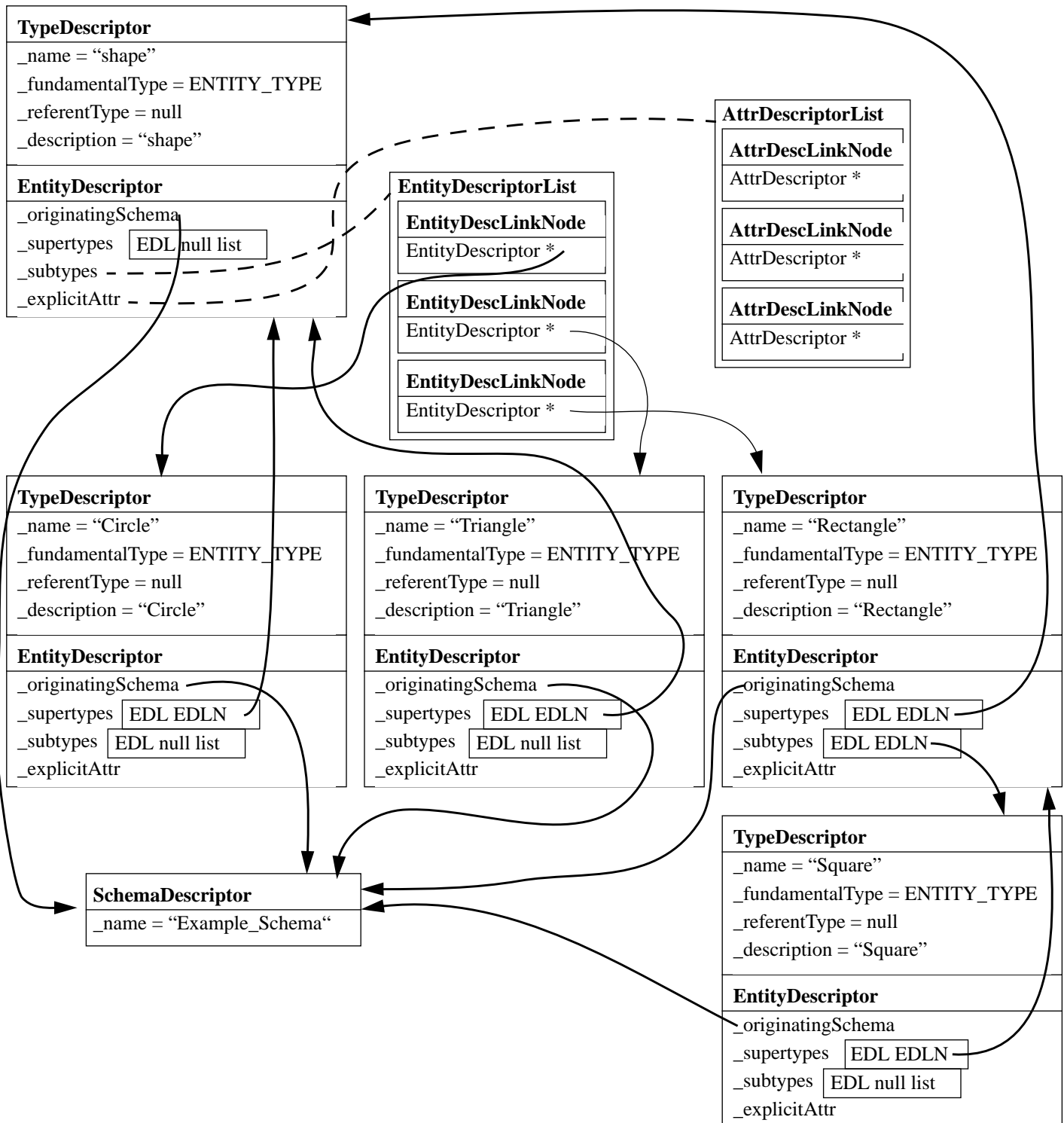
#### AttrDescLinkNode

AttrDescriptor \*

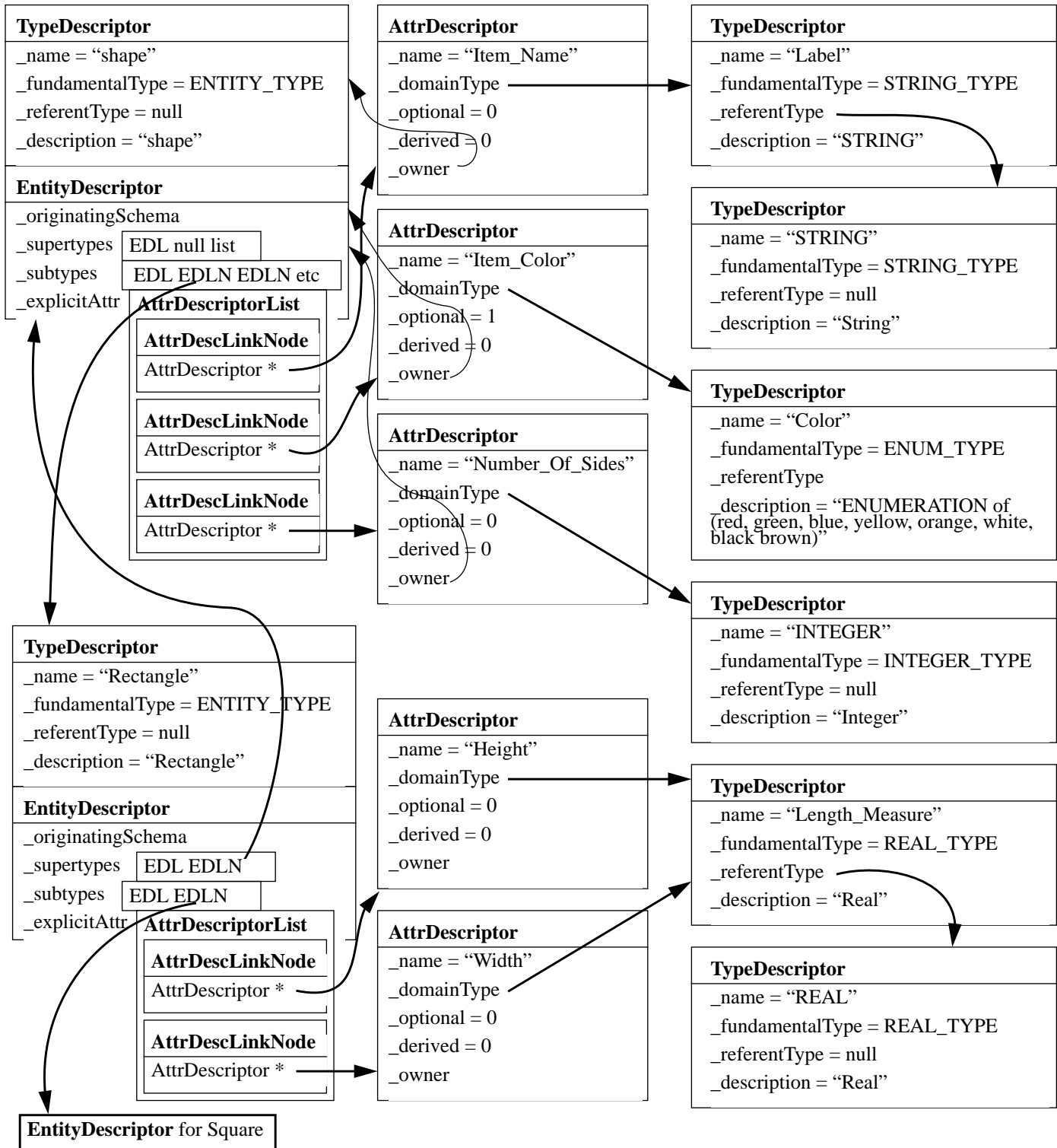
## Classes Used for Entity Instances



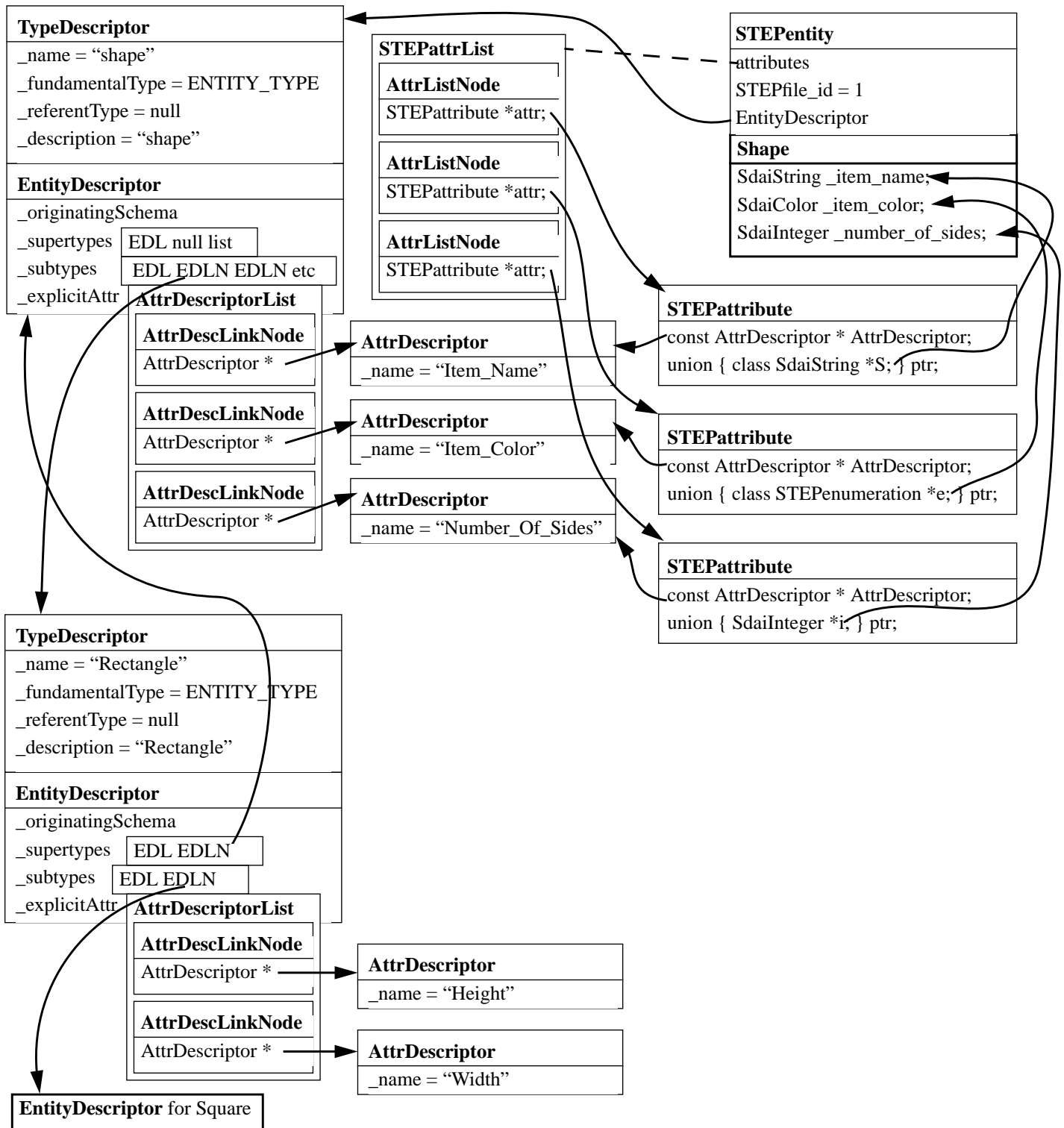
## Express Dictionary Classes



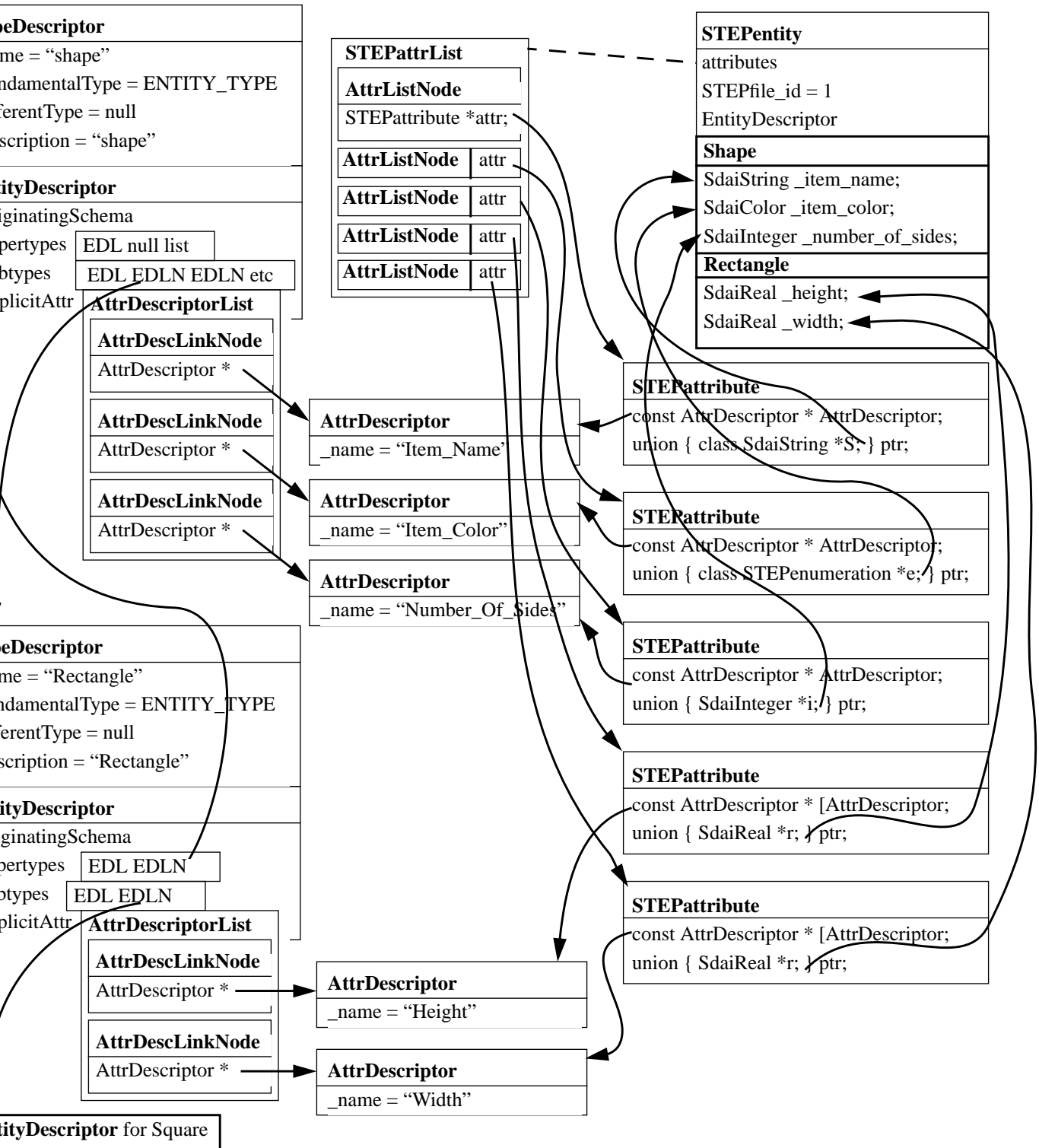
## Express Dictionary Classes



## Express Dictionary and Entity Instance Classes



## Express Dictionary and Entity Instance Classes



---

## Classes for STEPAttribute Instance Data Values

---

STEPentity  
STEPaggregate, STEPnode  
STEPenumeration  
SdaiSelect  
SdaiString  
SdaiBinary  
SCLundefined

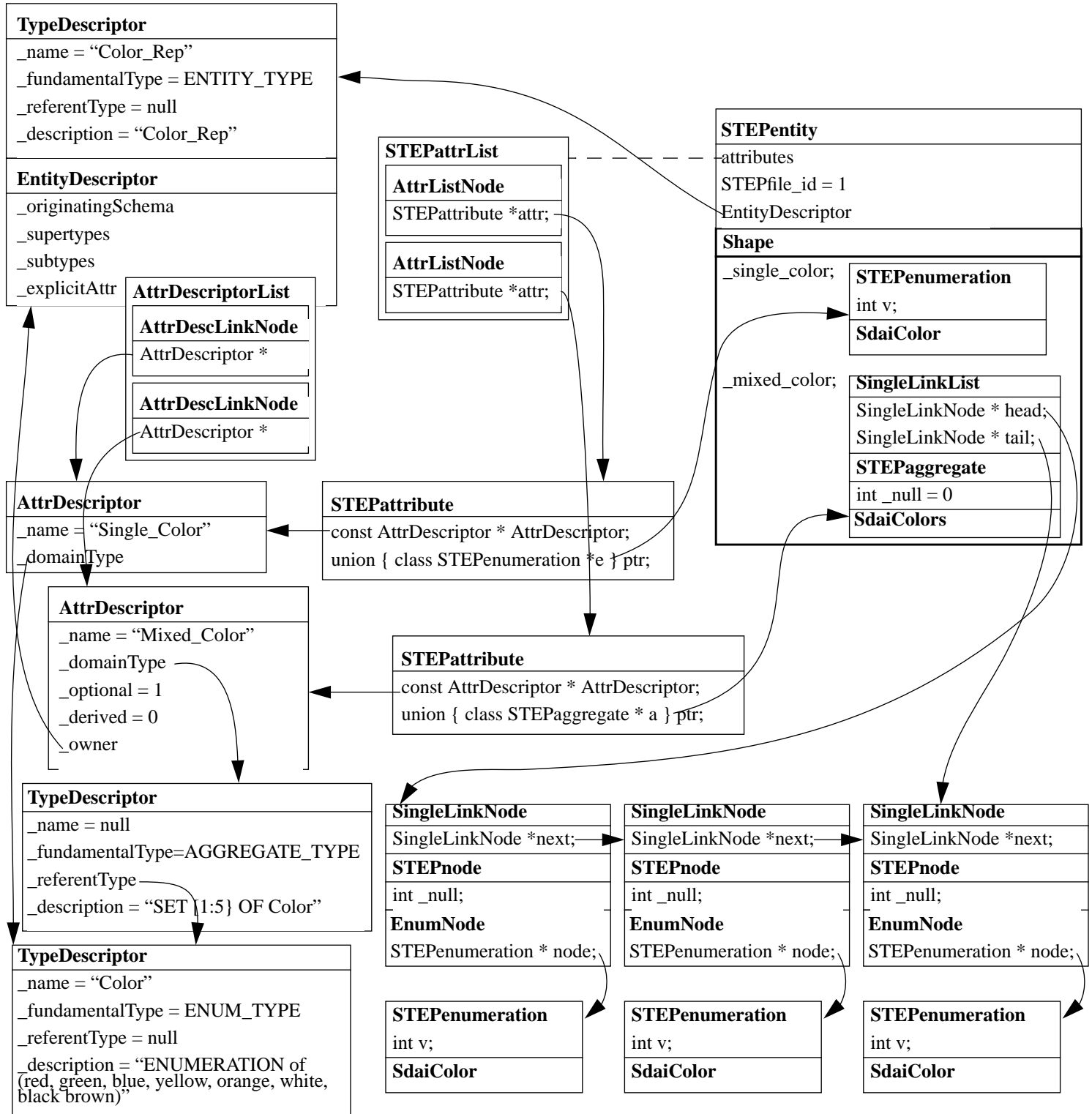
---

## Virtual Functions for the above classes

---

STEPread() - read Exchange File format  
STEPwrite() - write Exchange File format  
StrToVal() - read user interface format  
asStr() - write user interface format  
Validate functions

# Express Dictionary and Entity Instance Classes





## Express Used in These Notes

SCHEMA example\_schema;

TYPE label = STRING;  
END\_TYPE;

TYPE color = ENUMERATION OF (red, green, blue, yellow, orange, white, black, brown);  
END\_TYPE;

TYPE length\_measure = REAL;  
END\_TYPE;

TYPE point = REAL;  
END\_TYPE;

ENTITY color\_rep;  
    single\_color : OPTIONAL color;  
    mixed\_color : OPTIONAL set [1:5] of color;  
END\_ENTITY;

ENTITY shape  
    SUPERTYPE OF (ONEOF (circle, triangle, rectangle));  
    item\_name : label;  
    item\_color : OPTIONAL color;  
    number\_of\_sides : INTEGER;  
END\_ENTITY;

ENTITY circle  
    SUBTYPE OF (shape);  
    radius : real;  
END\_ENTITY;

ENTITY triangle  
    SUBTYPE OF (shape);  
    side1\_length, side2\_length, side3\_length : length\_measure;  
END\_ENTITY;

ENTITY rectangle  
    SUPERTYPE OF (square)  
    SUBTYPE OF (shape);  
    height : length\_measure;  
    width : length\_measure;  
END\_ENTITY;

## Express Used in These Notes (continued)

```
ENTITY square
SUBTYPE OF (rectangle);
END_ENTITY;
```

```
ENTITY cartesian_point;
  x : point;
  y : point;
  z : OPTIONAL point;
END_ENTITY;
```

```
ENTITY line;
  end_point_one : cartesian_point;
  end_point_two : cartesian_point;
END_ENTITY;
```

```
ENTITY poly_line;
  points : LIST OF line;
END_ENTITY;
```

```
END_SCHEMA;
```