基于分层匹配和最长公共子序列的SCD文件比较算法①

徐 睿, 陈宏君, 张 磊, 周 磊, 文继锋

(南京南瑞继保电气有限公司, 南京 211102)

摘 要: IEC61850 通信已经在电力系统中广泛使用, 其中变电站通信系统使用 SCD 文件进行描述. SCD 文件是 XML 格式的层次化结构, 不适合直接用文本按行对比来分析差异. 同时由于 SCD 文件层次结构多, 使用纯结构化的比较方法, 会导致比较结果冗长, 执行效率低. 本文基于 SCD 文件的特征, 提出了分层匹配的半结构化半文本比较思路. 先按照智能电子设备、连接接入点、逻辑设备等层次结构, 提取关键属性名, 进行对齐匹配. 之后在逻辑设备范围内, 针对逻辑节点的内容, 采用最长公共子序列的匹配算法对比局部文本内容, 该算法可去除仅调整顺序不影响实体内容的无效差异, 比较速度快, 比较结果准确直观.

关键词: IEC61850; SCD/ICD 文件; 层次比较; 最长公共子序列

SCD File Comparison Algorithm Based on Hierarchy Match and Longest Common Subsequence

XU Rui, CHEN Hong-Jun, ZHANG Lei, ZHOU Lei, WEN Ji-Feng

(NR Electric Co. Ltd., Nanjing 211102, China)

Abstract: IEC61850 communication has been widely used in electric power system, and SCD file is adopted to describe the Substation Communication System. Hierarchical structure of the SCD file is an XML format, making it not suitable for direct text comparison methods. Meanwhile, the levels of SCD file hierarchy are too many for pure structured comparison methods, which make them lack of time and space efficiency. Based on characteristics of the SCD file, this paper proposes a hierarchical combined approach of structured match and text comparison. Firstly, we abstract critical attribute names with hierarchy information of IED/AccessPoint/LDevice and compare the hierarchy correspondently with structured comparison method which is based on critical attributes. Secondly, we compare the content of LNode with what is based on longest common subsequence algorithm. This combined method could remove invalid differences arising by sequence modification, and could complete the comparison much faster with more accurate and straight forward comparison results.

Key words: IEC61850; SCD/ICD file; hierarchy match; longest common subsequence

IEC61850 通信已经在电力系统中被广泛使用, 其中 SCD(substation configuration description)文件和 ICD(IED capability description)文件在智能变电站的建设过程中起到了至关重要的作用. 由各个设备厂商提供 IED 设备的 ICD 文件, 由集成商实例化各个 IED 设备, 并且配置相关通信参数以及虚端子连接后, 形成全变电站的 SCD 文件[1,2]. 在整个配置建模的过程中, 由于需求的变化或功能的调整, SCD 和 ICD 文件也会发生变化. 因此需要能够方便的对比查看修改前后文件的差异, 便于工程调试和版本管理.

SCD 文件采用 XML 层次结构进行描述, 并且各个层次的节点都具有特定的语义, 如果使用纯文本的比较方法, 难以正确地、清晰地表示文件的差异[3-6]. 文献[3]提出了一种基于键/值的 SCL 文件比较方法, 即对关键节点定义键值和比较判据, 形成带主键的二维表结构, 比对文件的差异. 文献[4]提出了一种根据每个层次节点的属性遍历查找, 进行比对的方法. 这两种方法都是纯结构化的比较方法, 需要遍历查找整

① 基金项目:国家电网公司科技项目(DW1600052)

收稿时间:2016-03-26;收到修改稿时间:2016-05-05 [doi:10.15888/j.cnki.csa.005506]

个文件的层次结构. 由于 SCD 文件结构复杂, 节点层次多, 每个层次的节点所具有的属性都不同, 很难抽象出一种统一的比较模式. 因此结构化的比较方法效率会比较低, 并且对于比较结果的展示不是很直观.

本文提出一种半结构化半文本的比较方法. 首先使用结构化的比较方法比较 SCD 文件的主体结构, 即不比较所有层次的节点, 只比较 Header 节点、IED 节点、AcessPoint 节点、LDevice 节点等主干节点. 然后使用基于最长公共子序列的比较算法, 比较逻辑节点的内容. 计算最长公共子序列(Longest Common Subsequence, LCS)的算法在数据差异分析、文件版本控制、生物信息分析、图像处理等方面具有广泛的应用[7-9]. 本文将参与比较的两个逻辑节点的内容转换成两个序列, 通过计算这两个序列最长公共子序列实现对这两个序列的比较, 从而完成逻辑节点内容的比较. 使用这种半结构化半文本的比较方法既可以在保持文件主体结构不变的情况下去除仅调整顺序不影响实体内容的无效差异, 又可以快速准确地比较具体内容, 直观地展现比较结果.

1 SCD文件结构

SCD/ICD 文件采用 XML 层次化结构描述, 其结构如图 1 所示, 顶层结构包括: 文件头(Header)、通信配置(Communication)、智能电子设备(IED)、数据模板(DataTypeTemplates)[1].

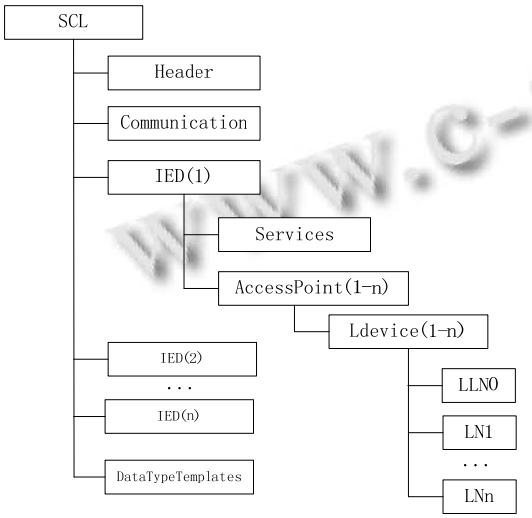

图 1 SCD 模型文件层次结构示例图

其中 IED 是 SCD 模型文件核心部分(可包括 1-N 个), IED 包括若干连接接入点(AccessPoint), 连接接入点包括若干逻辑设备(LDevice), 逻辑设备由 1 个公用逻辑节点(LLN0)、代表具体功能的若干逻辑节点(LN)组成. 逻辑节点由若干数据对象实例(DOI)组成, 数据对象由若干数据类的实例组合(DAI)组成.

在数据结构上, SCD 和 ICD 差异体现在 IED 的个数, 故 SCD 的比较算法也适用于 ICD 比较. 本文重点以 SCD 为例, 介绍实现方案.

2 层次化比较方法

对于两个 SCD 文件, 使用结构化分层向下的比较方法比较总体结构. 首先把文件按照层次结构进行分层. 然后对于同一层的节点, 以能够区分不同节点的内容作为关键字, 进行查找匹配. 如果查找到对应的节点, 则比较它们的内容以及它们的子节点. 比较子节点时, 同样以能够区分不同节点的内容作为关键字, 进行查找匹配, 这样一层层递归向下进行比较. 子节点的比较结果会影响父节点的比较结果.

SCD 文件的顶层节点主要包括四个: 文件头(Header)节点、通信配置(Communication)节点、智能电子设备(IED)节点和数据模板(DataTypeTemplates)节点. 总体比较流程如图 2 所示.

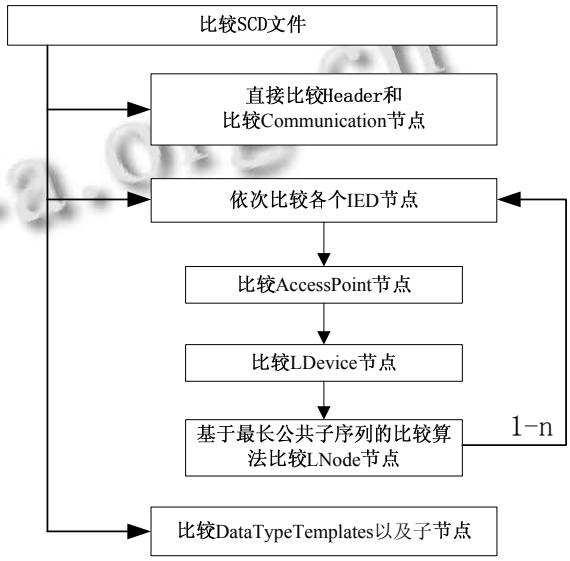

图 2 SCD 文件总体比较流程

对于 Header/Communication 等内容简单的节点, 直接比较它们的属性和内容, 就可以得到两个文件中对应的节点是否相同.

比较 IED 节点时, 以一个文件中的 IED 名称作为关键字, 在另一个文件中查找相同名称的 IED 节点. 如果有对应的节点, 则将这两个节点对齐, 并且比较属性和子节点(AccessPoint 节点). 如果它们的属性和子节点都相同, 则这两个 IED 节点被标记为相同. 如果没有对应的节点, 则在没有的文件中插入一个空节点.

比较 AccessPoint 节点时, 以一个文件中的接入点名称作为关键字, 在另一个文件中的匹配的 IED 节点下查找相同名称的 AccessPoint 节点. 如果有对应的节点, 则将这两个节点对齐, 并且比较属性和子节点(LDevice 节点). 如果它们的属性和子节点都相同, 则这两个 AccessPoint 节点被标记为相同. 如果没有对应的节点, 则在没有的文件中插入一个空节点.

文件 A　IED:PCS902			文件 B　IED:PCS902	
接入点名称	接入点属性及子节点		接入点名称	接入点属性及子节点
S1	…	相同	S1	…
M1	…	不同	M1	…
G1	…	没有对应节点		

图 3　比较 AccessPoint 节点

比较 LDevice 节点时, 以一个文件中的逻辑设备实例名作为关键字, 在另一个文件中匹配的 AccessPoint 节点下查找相同实例名的 LDevice 节点. 如果有对应的节点, 则将这两个节点对齐, 并且比较它们的属性和子节点(LN 节点). 如果它们的属性和子节点都相同, 则这两个 LDevice 节点被标记为相同. 如果没有对应的节点, 则在没有的文件中插入一个空节点.

文件 A　AccessPoint:S1			文件 B　AccessPoint:S1	
逻辑设备实例名	逻辑设备属性及子节点		逻辑设备实例名	逻辑设备属性及子节点
LD0	…	不同	LD0	…
PROT	…	不同	PROT	…
RCD	…	相同	RCD	…
		没有对应节点	CTRL	…
…	…		…	…

图 4　比较 LDevice 节点

比较 LNode 节点时, 以一个文件中的逻辑节点的类型+实例号作为关键字, 在另一个文件中对应的 LDevice 节点下查找具有相同类型+实例号的 LNode 节点. 如果有对应的节点, 则将这两个节点对齐, 并且比较它们的属性和内容, 使用下一节介绍的方法进行比较. 如果它们的属性和内容都相同, 则这两个 LNode 节点被标记为相同. 如果没有对应的节点, 则在没有的文件中插入一个空节点.

文件 A　LDevice: PROT			文件 B　LDevice: PROT	
逻辑节点类型+实例号	逻辑节点属性及内容		逻辑节点类型+实例号	逻辑节点属性及内容
LLN0	…	不同	LLN0	…
LPHD1	…	相同	LPHD1	…
		没有对应节点	LPHD2	…
PDIS1	…	相同	PDIS1	…
…	…		…	…

图 5　比较 LNode 节点

对于 DataTypeTemplates 节点, 分别比较它的四个子节点(LNodeType 节点、DOType 节点、DAType 节点和 EnumType 节点)的内容. 由于这四个子节点的层次不是很深, 所以仍然使用层次化的比较方法. 比较时分别使用各个节点 ID 属性作为关键字进行查找匹配, 如果查找到对应的节点, 则进一步比较它们的属性和内容.

3　基于最长公共子序列的逻辑节点比较算法

由于逻辑节点(LN)由多个 DO 节点组成, DO 节点由多个 DA 节点组成, DA 有由多个基本数据类型节点组成, 并且每一层节点都是使用 XML 文件格式定义, 层次结构较多. 如果仍然使用结构化的方法比较, 比较结果会显得冗长, 比较效率会比较低. 因此使用基于最长公共子序列的文本化比较算法来比较逻辑节点的内容. 首先给出序列、子序列和最长公共子序列的定义.

3.1 概念定义

定义 1. 序列: 表示排列成一列的元素列表, 元素之间相对位置是确定的[9].

定义 2. 子序列: 由一个序列通过去除某些元素但不破坏余下元素的相对位置而形成的新序列[9].

如果序列 A 通过删除零个或多个元素可以得到序列 B, 那么 B 就是 A 的一个子序列.

子序列与子串的的相同之处: 无论 B 是 A 的子序列还是子串, A 和 B 中对应的元素出现的顺序是一致的.

子序列与子串的区别: 如果 B 是 A 的子串, 那么 B 中的元素在 A 中是连续出现的; 而如果 B 是 A 的子序列, 那么 B 中的元素在 A 中可以不连续出现.

例如: 对于序列“compare”, “par”是一个子串, 同时也是一个子序列, 而“opre”只是一个子序列.

定义 3. 最长公共子序列(LCS): 序列 A 和序列 B 的最长公共子序列就是在两个序列中都出现的子序列中的最长的一个子序列[9].

比较两个逻辑节点的内容时, 实际上是比较两段文本的内容. 可以把每一段文本类比成一个序列, 文本中的每一行类比成一个元素. 那么比较两个逻辑节点的问题就可以转化为比较两个序列的问题. 比较两段文本时希望找出两段文本中最多的匹配部分, 也就是要找出两个序列中最多的匹配部分. 因此可以通过计算两个序列的最长公共子序列(LCS), 来找到这两个序列的最多匹配部分.

定义 4. LCS(A, B): 表示序列 A 和序列 B 的最长公共子序列的长度.

假设序列 $A = a_1a_2……a_M$, 即 A 是由 $a_1a_2……a_M$ 这 M 个元素顺序组成. $a_1a_2……a_i$ 表示 A 的前 i 个元素组成的子序列.

假设序列 $B = b_1b_2……b_N$, 即 B 是由 $b_1b_2……b_N$ 这 N 个元素顺序组成. $b_1b_2……b_j$ 表示 B 的前 j 个元素组成的子序列.

定义 5. LCS(i, j): 表示序列 A 的前 i 个元素组成的子序列 $a_1a_2……a_i$ 和序列 B 的前 j 个元素组成的子序列 $b_1b_2……b_j$ 的最长公共子序列的长度.

即 $LCS(i, j) = LCS(a_1a_2……a_i, b_1b_2……b_j)$, 其中 $1 \le i \le M, 1 \le j \le N$.

那么 $LCS(M, N) = LCS(a_1a_2……a_M, b_1b_2……b_N) = LCS(A, B)$.

3.2 计算最长公共子序列

假设序列 A 和序列 B 的最长公共子序列为 $S = s_1s_2……s_K$

若 $a_M = b_N$,

则 $s_K = a_M = b_N$, 且 $s_1s_2……s_{K-1}$ 是 $a_1a_2……a_{M-1}$ 和 $b_1b_2……b_{N-1}$ 的最长公共子序列.

此时 LCS(M, N) = LCS(M-1, N-1)+1

若 $a_M \ne b_N$, $s_K \ne a_M$, $s_K = b_N$,

则 $s_1s_2……s_K$ 是 $a_1a_2……a_{M-1}$ 和 $b_1b_2……b_N$ 的最长公共子序列.

此时 LCS(M, N) = LCS(M-1, N)

若 $a_M \ne b_N$, 且 $s_K = a_M$, $s_K \ne b_N$,

则 $s_1s_2……s_K$ 是 $a_1a_2……a_M$ 和 $b_1b_2……b_{N-1}$ 的最长公共子序列.

此时 LCS(M, N) = LCS(M, N-1)

若 $a_M \ne b_N$, 且 $s_K \ne a_M$, $s_K \ne b_N$,

则 $s_1s_2……s_K$ 是 $a_1a_2……a_{M-1}$ 和 $b_1b_2……b_{N-1}$ 的最长公共子序列.

此时 LCS(M, N) = LCS(M-1, N-1)

由数学归纳法的思想可以得到如下计算公式:

<公式一> 对于 $1 \le i \le N, 1 \le j \le M$,

若 $a_i = b_j$, 则 $LCS(i, j) = LCS(i-1, j-1) + 1$

若 $a_i \ne b_j$, 则 $LCS(i, j) = Max(LCS(i-1, j-1), LCS(i-1, j), LCS(i, j-1))$

由以上计算方法可以得出: 序列 A 和序列 B 的最长公共子序列是由 A 的子序列和 B 的子序列的最长公共子序列所决定的. 因此需要计算 A 的所有子序列和 B 的所有子序列之间一一对应的最长公共子序列, 这样才能获得 A 和 B 的最长公共子序列.

3.3 计算匹配矩阵

对于序列 A 和序列 B, 定义一个 M*N 的匹配矩阵来计算 A 和 B 中所有子序列的最长公共子序列的长度. 矩阵中第 i 行、第 j 列的元素值即表示 A 的子序列 $a_1a_2……a_i$ 和 B 的子序列 $b_1b_2……b_j$ 的最长公共子序列的长度.

假设序列 A = GGATCGA, 序列 B = GATTCAGTTA. 定义匹配矩阵, 并将矩阵中的第 0 列和第 0 行的元素值初始化为 0, 然后从矩阵的左上角到右下角, 根据<公式一>依次计算第 1 行第 1 列至第 7 行第 11 列的各个元素值, 计算结果如图 6 所示.

		0	1	2	3	4	5	6	7	8	9	10
			G	A	T	T	C	A	G	T	T	A
0		0	0	0	0	0	0	0	0	0	0	0
1	G	0	1	1	1	1	1	1	1	1	1	1
2	G	0	1	1	1	1	1	1	2	2	2	2
3	A	0	1	2	2	2	2	2	2	2	2	3
4	T	0	1	2	3	3	3	3	3	3	3	3
5	C	0	1	2	3	3	4	4	4	4	4	4
6	G	0	1	2	3	3	4	4	5	5	5	5
7	A	0	1	2	3	3	4	5	5	5	5	6

图 6 计算匹配矩阵

3.4 计算回溯路径

计算得到两个序列的匹配矩阵后, 再从矩阵的右下角往左上角进行回溯. 假设当前位于矩阵的第 i 行

第 j 列，则根据<公式一>得到的回溯规则如下:

若 $a_i = b_j$，则回溯到当前元素的左上角元素.

若 $a_i \neq b_j$，则回溯到当前元素的左上角元素、上边元素和左边元素中值最大的一个. 若存在相等的情况，则可以取其中的任意一个.

若当前元素位于矩阵的第一行，则回溯到当前元素的左边元素.

若当前元素位于矩阵的第一列，则回溯到当前元素的上边元素.

依据回溯规则对匹配矩阵进行回溯，得到一条回溯路径如图 7 所示.

		0	1	2	3	4	5	6	7	8	9	10
			G	A	T	T	C	A	G	T	T	A
0		0	0	0	0	0	0	0	0	0	0	0
1	G	0	1	1	1	1	1	1	1	1	1	1
2	G	0	1	1	1	1	1	2	2	2	2	2
3	A	0	1	2	2	2	2	2	2	2	2	3
4	T	0	1	2	3	3	3	3	3	3	3	3
5	C	0	1	2	3	3	4	4	4	4	4	4
6	G	0	1	2	3	3	4	4	5	5	5	5
7	A	0	1	2	3	3	4	5	5	5	5	6

图 7　回溯匹配矩阵

3.5 根据回溯路径获得最优匹配

根据回溯路径，可以计算得到序列 A 和序列 B 的分别对应的具有最多公共部分的匹配序列 A’和 B’. 序列 A’和 B’表示了原始序列 A 和 B 的一个最优匹配，即在对应位置上具有最多相同内容.

沿着回溯路径(图 7 中黄色的部分)从右下角往左上角移动，假设当前元素位于矩阵的第 i 行第 j 列.

如果回溯路径上的下一个元素在当前元素的左上角(第 i-1 行第 j-1 列)，则将 a_{i-1} 添加到 A’的开始位置，将 b_{j-1} 添加到 B’ 的开始位置.

如果回溯路径上的下一个元素在当前元素的左边(第 i 行第 j-1 列)，则将一个空元素添加到 A’ 的开始位置，将 b_{j-1} 添加到 B’ 的开始位置.

如果回溯路径上的下一个元素在当前元素的上边(第 i-1 行第 j 列)，则将 a_{i-1} 添加到 A’ 的开始位置，将一个空元素添加到 B’ 的开始位置.

按照上述方法，沿着回溯路径从右下角回溯到左上角后，就可以得到 A’和 B’，分别如下所示，其中“_”表示一个空元素. 通过序列 A’和 B’可以直观地看出原始序列 A 和 B 的相同部分(并且是最多的相同部分)和不同部分.

A’: GGA_TC_G__A

B’: _GATTCAGTTA

这也是比较两个逻辑节点的内容时，所希望得到的结果. 把逻辑节点的文本内容转化为一个序列，即文本中的每一行表示一个元素，如图 8 所示.

逻辑节点文本	序列 B
<DOI name="FuncEna1" desc="Function 1 enabled">	G
<DAI name="ctlModel">	A
<Val>0</Val>	T
</DAI>	T
<DAI name="sboTimeout">	C
<Val>30000</Val>	A
</DAI>	G
<DAI name="dataNs">	T
<Val>SGCC MODEL:2012</Val>	T
</DAI>	A

图 8　逻辑节点的文本内容转化为序列

然后使用上述基于最长公共子序列的比较算法，可以得到两个逻辑节点内容的比较结果，即可以得到两个逻辑节点中最多的相同部分，又可以显示出不同的部分，并且保留了原有逻辑节点内容的顺序，进行对齐匹配，处理流程如图 9 所示.

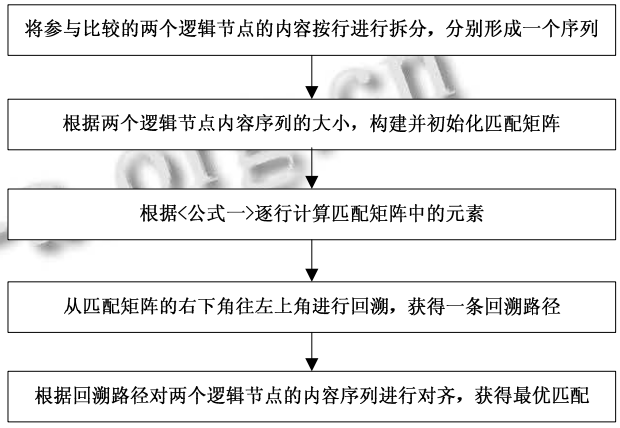

图 9　逻辑节点内容比较流程

4　算法验证

基于这种结构化比较方法和计算最长公共子序列的文本比较方法相结合的比较方法，实现了一个 SCD 文件的比较工具. 工具可以对两个 SCD 文件进行比较，并且显示总体结构的比较结果，以及各个节点的详细比较结果，见附录 A. 对于逻辑节点等结构比较复杂，嵌套层次比较多的节点，使用基于最长公共子序列的

文本比较算法进行比较, 比较结果见附录 B.

5 结语

本文提出了一种分层的结构化比较和基于最长公共子序列的文本比较相结合的比较算法, 实现了 SCD/ICD 文件的差异比较和展示. 使用此算法实现的 SCD/ICD 文件比较功能已经在保护测控装置配套软件 PCS-Explorer 中使用[10], 已经广泛应用于智能变电站的工程集成中.

参考文献

1 IEC/TC57. Communication networks and systems for power utility automation-Part 6: Configuration description language for communication in electrical substation related to IEDs. Ed 2.0. 2009.

2 智能变电站调试规范.国家电网公司企业标准,Q/GDW 689-2012,2012.

3 高磊.IEC61850SCL 配置文件比对工具的研究与实现.电力系统自动化,2013,20(10):88–91.

4 马杰,李磊,黄德斌,等.智能变电站二次系统全过程管控平台研究与实践.电力系统保护与控制,2013,41(2):67–72.

5 樊陈,倪益民,窦仁晖,等.智能变电站信息模型的讨论.电力系统自动化,2012,36(13):15–19.

6 笃峻,叶翔,王长瑞,等.智能变电站设计配置一体化功能规范研究及工具开发.电力系统自动化,2014,38(20):85–89.

7 Bergroth L, Hakonen H, Raita T. A survey of longest common subsequence algorithms. International Symposium on String Processing & Information Retrieval. 2000. 39–48.

8 曾波,潘少彬,陆璐,等.改进的 LCS 方法在测试脚本序列比对中的应用.计算机工程与应用,2011,47(35):71–76.

9 Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms (2nd ed.). MIT Press and McGraw-Hill, 2001.

10 陈宏君,刘克金,冯亚东,等.新一代保护测控装置配套工具软件设计与应用.电力系统自动化,2013,37(20):92–96.