和配制酱油仅通过该指标很难区分。颜勇等分析了国家标准提出利用氨基酸态氮与全氮比值来区分酿造，配制和化学酱油，氨基酸态氮与全氮比值在0.50--0.57范围内，为酿造酱油，氨基酸态氮与全氮比值在 0.57--0.64 范围内为配制酱油，氨基酸态氮与全氮比值在 0.64--0.79 范围内，为化学酱油。而张瑾等研究发现， 由于焦糖的全氮含量较高，氨基酸态氮含量较少，如果添加大量焦糖，酱油的全氮也就会增长的较多，这样就使氨基酸态氮变化比较小。这使得颜勇等人的鉴别方法不能很好的鉴别真假酿造酱油。还有鉴别方法根据酱油中氯丙醇或乙酰丙酸有无来鉴别为配制酱油还是酿造酱油。对于氯丙醇的测定主要是利用色谱学方法，对于乙酰丙酸测定主要有化学显色、纸色谱和气相色谱方法，化学显色方法需要把乙酰丙酸经过萃取浓缩，再与香草醛、硫酸显色，操作繁琐，很难在现场快速实施。纸色谱和气相色谱方法也只能在实验室进行，不能现场快速检测。

本课题采用气质联用技术筛选出酿造酱油中都有的乙醇和色氨酸，针对乙醇和色氨酸的特异显色反应，优化检测体系，构建稳定快速检测试剂盒。

3.2 脱色剂的选择

本课题组尝试了应用漂泊土、脱色砂、二氧化硅、硅藻土、沸石、活性炭等做吸附脱色剂，对酱油样品进行脱色实验。二氧化硅脱色剂具有物理吸附作用强、化学稳定性等优点，但对水溶性色素吸附效果差 , 对酱油脱色效果差。漂泊土絮凝沉降能力较弱、对酱油脱色效果弱，而且价格高。硅藻土脱色效果依赖其多孔结构，对某些复杂色素或高粘度液体（如酱油）的吸附能力有限，对酱油脱色效果不理想。活性炭是一种内部孔隙结构发达、比表面积大、吸附性能强，有比较好的吸附作用，是一种优良的吸附剂 , 价格便宜 , 是对酱油进行脱色的最好选择。本课题组利用活性炭对酱油进行脱色，经多次试验，制作的脱色装置对酱油脱色效果很好，脱色后样品清彻透明，可以进行下一步的比色实验。脱色装置制作简单，价格便宜，可以现场操作简单快速。而且，经过试验表明，活性炭对酱油中的乙醇和色氨酸没有吸附，所以本课题组选择活性炭作为酱油的脱色剂。

3.3 验证试验

根据相关文献报道课题组配制了三种化学酱油：

酱色、食盐、味精和水四种配制的简单的一号化学酱油

色、食盐、味精、水、酱油助香剂、增稠剂、高效防腐剂调配的二号化学酱油

酱色、普通食用盐、水和蛋白水解液配制的三号化学酱油

以上述三种化学酱油和六种酿造酱油组成盲样，请课题组其他不知道盲样结果的同志用本方法进行定性试验，结果如下表 1：

上述验证实验结果与真实结果完全一致，没有出现假阳性和假阴性的结果，表明本方法准确可靠。

表 1 酿造酱油与化学酱油的鉴别验证试验

编号	乙醇检测	色氨酸检测	实验结果	真实结果
1	紫色	淡黄色	酿造酱油	李锦记锦珍生抽
2	紫色	淡黄色	酿造酱油	味事达金标生抽王
3	不显色	不显色	化学酱油	二号化学酱油
4	紫色	淡黄色	酿造酱油	海天金标生抽
5	不显色	不显色	化学酱油	三号化学酱油
6	紫色	淡黄色	酿造酱油	东古一品鲜
7	紫色	淡黄色	酿造酱油	厨邦小淘气
8	不显色	不显色	化学酱油	一号化学酱油
9	紫色	淡黄色	酿造酱油	乐购美月鲜

3.4 试剂盒的保存时间试验

制作一批试剂盒分别按两组保存条件保存：第一组是塑料瓶中、避光、室温下保存，第二组是塑料瓶中、避光、冰箱中 4℃保存。两组测试的时间间隔为 第 7 天 , 第 28 天 , 第 56 天 , 第 90 天、以后每隔 3 个月做一次。实验表明，试剂盒在室温下保质期为 12 个月，冰箱中 4℃保质期 12 月以上。所以试剂盒试剂保存简单，稳定可靠。

3.5 问题讨论

本试剂盒还不能鉴别利用酿造酱油和化学酱油混合勾兑的配制酱油，因为市场上不同品牌产品所含乙醇和色氨酸含量不同，所以不能根据乙醇和色氨酸含量少就认定为化学酱油。如果是其它的化学酱油，本试剂盒就能现场快速鉴定。

4 结语

本试剂盒研制完成后，请了三家检测机构鉴定，鉴定意见认为本课题用液质连用技术筛选酿造酱油中特有成分，针对这些特有成分的特异显色反应，优化检测体系，构建了快速检测试剂盒，国内未见报道。本试剂盒操作方便、可现场快速定性检测，对现场进行酱油打假很有帮助。

参考文献

[1] THOMAS R, ANT JE P, HERBERT S, et a .l The chem istry of side reactions and byproduct formation in the systemNMMO / ce llulose(Lyoce ll process) [J]. Prog Polym Sc,i 2001, 26: 1763-1837.

[2] 王林祥, 刘杨岷, 王建新. 酱油风味成分的分离鉴定[J]. 中国调味品, 2005, 1: 45- 48

[3] LEE SM, SEO B C, KIM Youngsuk. Vola tile compounds in femented and acid- hydrolyzed soy sauce[J] . J Food Sci 2006, 71(3): 146- 156

[4] 冯笑军, 吴惠勤, 黄晓兰，等.气相色谱- 质谱对天然酿造酱油与配制酱油香气成分的分析比较[J].分析测试学报，2009，28(6)：661-665

[5] Wanakhachornkrai P, Lertsiri S.Comparison of determination methodforvolatile compounds inThai soysauce [J].FoodChemistry, 2003(83) :619- 629

 DOI: https://doi.org/10.12349/tie.v2i11.8701

Analysis of ASTERIX data format defects and thoughts on China's independent monitoring data format

Bin Long Wenjuan Hu Xiaoyue Jiang

Chongqing Air Traffic Control Branch, Civil Aviation Administration of China, Chongqing 401120, China

Abstract

This paper conducts an in-depth analysis of the ASTERIX protocol, revealing its significant shortcomings in modern air traffic control environments. It systematically examines issues such as the lack of unified standards for data structures, unreasonable design, and complex decoding, and proposes improvement suggestions for the ASTERIX protocol. Based on a thorough analysis of the ASTERIX protocol's flaws, this paper further proposes the direction for constructing China's independent air traffic control data format, aiming to provide theoretical references and technical support for developing more efficient and standardized domestic air traffic control monitoring data exchange standards.

Keywords

ASTERIX; data format; defect; analysis; air traffic control; monitoring; autonomous.

ASTERIX 数据格式缺陷分析及中国自主监视数据格式的思考

龙滨 胡文娟 江晓玥

民航重庆空管分局，中国·重庆 401120

摘 要

本文针对ASTERIX协议进行了深入分析，揭示了该协议在现代空管环境中存在的显著缺陷，系统性地剖析了该协议在数据结构缺乏统一标准、设计不合理、解码复杂等问题，并提出了ASTERIX协议改进建议。在深入分析ASTERIX协议缺陷的基础上，本文进一步提出了构建中国自主空管数据格式的思考方向，旨在为发展更加高效、规范的国产空管监视数据交换标准提供理论参考和技术支撑。

关键词

ASTERIX；数据格式；缺陷；分析；空管；监视；自主

1 引言

ASTERIX 协议，也称通用结构化欧控监视信息交换协议，为欧控组织所开发，在空管监视数据中广泛使用。笔者在实现 ASTERIX 数据格式的通用解码算法过程中，积累了丰富的解码经验。然而在实际解码过程中，发现 ASTERIX 在数据设计和使用方面上存在诸多问题，这些问题在一定程度上影响了其在现代空管环境中的性能表现。

ASTERIX 数据格式于 1986 年正式发布，至今已有 39 年的历史。该数据格式在当时是非常先进，一是最大限度地减少传输数据量，在当时数据传输率受限的情况下，通过精心设计，合理利用每个字节甚至每个位的信息，实现数据的紧凑编码；二是在设计上考虑了前向兼容和后向兼容能力，FSPEC 的设计使得其数据结构可以不断扩展，为系统的升级和扩展提供了便利。

该协议在现今环境下，却呈现出两个明显的缺陷：一是数据解码处理开销大，由于其数据结构复杂，解码过程需要消耗较多计算资源并产生延迟，这对于实时性要求较高的空管系统，无疑造成了性能瓶颈；二是数据格式不统一，不同的 CAT 工作组在定义数据格式时较为随意，缺乏统一的标准和规范，导致不同类型数据之间的格式差异较大，增加了解码的复杂性和难度，通常解码人员需要为每个 CAT 类设计一套解码算法。

本文将从欧控文档定义到解码角度，对 ASTERIX 数据格式展开详细分析，深入挖掘出其存在的缺陷与改进方向，并在此基础上，提出关于构建中国自主空管数据格式的思考。

【作者简介】龙滨（1976–），男，中国重庆人，硕士，高级工程师，从事空中交通管理通信、导航、监视研究。

2 ASTERIX 数据格式的缺陷分析

数据格式总体结构设计不合理：在《欧控规范监视数据交换 - 第 1 部分》（2.2 版以上），对数据块（data block）的定义如下：

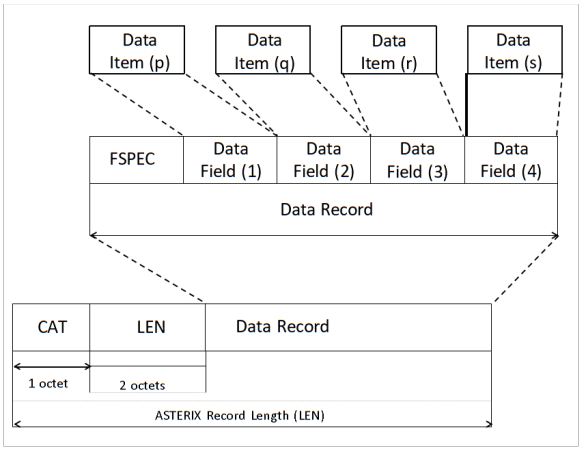

图 1 ASTERIX 数据总体结构

从层级结构上来看，其数据分为三个大类：类别（CAT）、数据长度（LEN）和数据记录（Data Record）。其中，数据记录又细分为字段说明（FSPEC）和多个数据字段（Data field），而每个数据字段对应每个数据项（Data Item）。在每个 CAT 数据文档中，用户应用配置（UAP）对每一个数据项进行了定义，其中包括保留扩展字段（REF）和保留特定用途 (SP) 字段。值得注意的是，保留扩展字段的定义同普通 DATA ITEM 的定义存在较大差异，且是单独成文的。故笔者认为 ASTERIX 数据格式总体结构存在的问题如下：①概念划分过于繁杂。将数据记录、数据字段和数据项分成三个不同的概念，一在定程度上增加了理解和使用难度。笔者更建议只需要用数据项的概念，其中的 CAT、LEN、FSPEC、DATA ITEM 均可看做数据项。②树形对象结构未充分体现。ASTERIX 格式中实际上存在树形对象结构的概念，其 1+1+ 类型用于将数据扩展到下一层级。然而，在总体结构图中，这一重要的结构特征并未得到体现，使得整个数据结构的完整性和直观性受到一定影响。③ REF 定义必要性存疑。REF 的设计初衷是为给定的类别进行中间变更提供一种机制，但在实际应用场景中，其必要性并不显著，完全可以在新增类别时，通过在常规数据项上直接新增数据项的方式实现类似的功能。

数据分类不清晰：在《欧控规范监视数据交换 - 第 1 部分》，对数据项结构有几种定义：①固定长度数据项（Fixed Length Data Items）②扩展长度数据项（Extended Length Data Items) ③显式长度数据项（Explicit Length Data Items）④重复数据项（Repetitive Data Items）⑤组合数据项（Compound Data Items）。以 cat048 为例，在其 UAP 文档中（如表 1 所示），数据项结构既有固定长度 (比如 2、3) 等形式，又存在 1+、1+8*n 以及 1+1+ 类型等特殊形式。这种多样的定义方式本应是为了更灵活地满足各种应用场景的需求，但实际情况却并非如此。

FRN	Data Item	Data Item Description	Length in Octets
1	I048/010	Data Source Identifier	2
2	I048/140	Time-of-Day	3
3	I048/020	Type and Properties of the Target Report and Target Capabilities	1+
4	I048/040	Measured Position in Slant Polar Coordinates	4
5	I048/070	Mode-3/A Code in Octal Representation	2
6	I048/090	Flight Level in Binary Representation	2
7	I048/130	Radar Plot Characteristics	1+1+
FX	n.a.	Field Extension Indicator	n.a.
8	I048/220	Aircraft Address	3
9	I048/240	Aircraft Identification	6
10	I048/250	Mode S MB Data	1+8*n

表 1 CAT048 UAP 部分数据项定义

问题 1：什么是 1+ 类型、和 1+1+ 类型?

在欧控文档的第一部分并未对这些特殊类型详细说明，这就导致后续 CAT 工作组在使用过程中缺乏明确的指导，产生大量不规范使用的案例。例如，从实际应用分析，“1+”类型应该是按照单个字节进行 FX 扩展，那“1+1+”类型又是什么？以上图的 Radar Plot Charecteristics 的数据格式分析，发现它类似 FSPEC 的字段定义，而在 cat004 中的 Aircraft Identification & Characteristics 1 存在将“1+1+”误用为“1+”的情况。这种误用情况屡屡出现，暴露出不同 CAT 工作组对这两种类型的区分并不清晰，严重影响了数据的一致性。

问题 2：1+1+ 类型下层的数据在 UAP 中是没有定义的，这是为什么？

通常情况下，不同层级的数据应该具有相对平等的地位，并且都有明确的类型和长度等信息。然而，在这里只有最上层的 FSPEC 指定的数据项有类型，“1+1+”再往下层的字段却没有数据项编号和长度信息，这给数据的解码和结构理解带来了极大的困难。

问题 3：那有 1+1+ 类型，是不是也就应该有 1+1+1+ 类型，甚至更深层次?

在文档中对此并未给出任何说明。虽然从理论上讲，有“1+1+”下一层的设计，就应该具备扩展到更多层设计的可能性，但由于缺乏明确规定，使得笔者在进行数据设计和处理时感到困惑。另外，“1+1+”中的这两个 1 到底是代表什么。遗憾的是，官方也未给出说明。

问题 4：1+ 还代表重复?

“1+”在实际应用中，有两种用法。例如在 CAT001 的 Target Report Descriptor 中，共定义了 2 个字节的数据格式，每个字节有不同的数据格式；而在 CAT048 的 Warning/Error Conditions 字段，只定义了 1 个字节，且每个字节低位为 FX，推测应该是这个字节只要 FX 不为 0，就可以不断循环。这里的问题在于，如果“1+”代表重复用途，那这种数据格式不就是《欧控规范监视数据交换 - 第 1 部分》规

定的重复数据项吗？如果是重复数据项，是否采用“1+1*n”的数据格式可能会更加合理一些？尽管会增加了一个字节，但却能提高数据的可读性，提升规范性。

问题 5：有没有 2+、3+ 之类的数据项?

在 CAT008 中的 process status 项定义中出现了“3+”类型的数据项（如图 2），同“1+”类型一样，其定义是否合理值得商榷。从提高可读性的角度出发，将其设计为“1+3*n”似乎更为合理，这样不仅保留了原有的功能，还能让数据的结构和含义更加清晰明了。

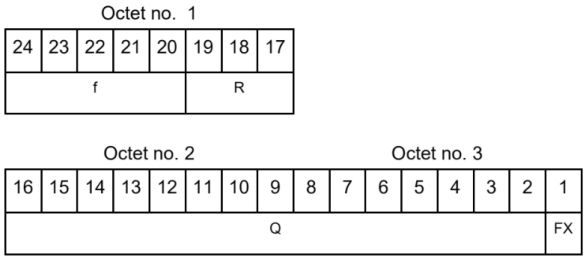

图 2 CAT008 的 process status 项定义

问题 6：有 1+ 和 2+ 的定义，那有没有不规则定义的数据项?

除了上述常见的几种类型外，还存在不规则定义的数据项。例如，在 CAT048 的保留扩展字段中（如图 3 所示），SGV 定义的第一个字段是 2 个字节，而扩展的第二个字段是 1 个字节。这种情况下，为什么不采用“2+”类型来定义，而要创造一种新的数据类型呢？这不仅增加了数据理解和处理的难度，也破坏了整个数据格式体系的一致性。

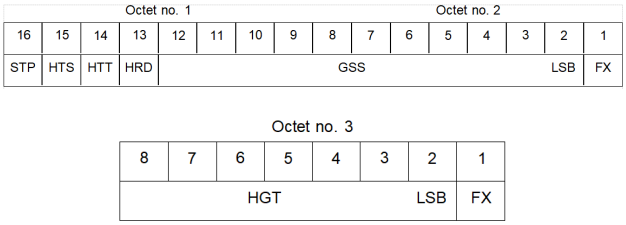

图 3 CAT048 保留扩展字段 SGV 定义

问题 7：重复数据项的 rep 字节数等于 1，能否大于 1？

从已知的资料上来看，尚未发现有这种情况，但从解码角度考虑，最好还是考虑 rep 字节数大于 1 的情况。

问题 8：保留扩展字段有 FSPEC 定义吗?

以 CAT048 和 CAT021 的保留扩展字段为例（如图 4、图 5 所示），它们的 FSPEC 定义存在问题。在这些例子中，FSPEC 的最后 1 位不是 FX，而是占满了。这与 CAT 类的 FSPEC 数据定义保留不一致，使得数据的扩展性受到限制。笔者更建议在保留扩展字段保留 FX 的使用，使得与 FSPEC 的用法保持一致，更方便用户对数据的理解和解码处理。

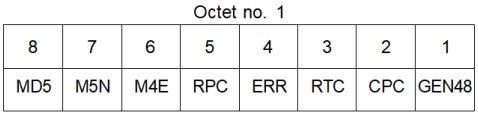

图 4 CAT048 保留扩展字段定义

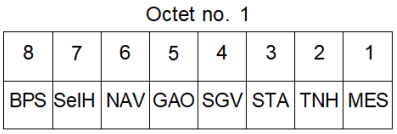

图 5 CAT021 保留扩展字段定义

问题 9：保留扩展字段的长度为什么不是 2 个字节?

所有的保留扩展字段的 LEN 都只有 1 个字节，这意味着其最大长度只能达到 255 个字节。相比之下，CAT 类的 LEN 占 2 字节，共有 65535 字节，可以传输更长的数据。这种差异导致了保留扩展字段的扩展能力受到了极大的限制，无法满足一些数据量需要较大的场景使用。此外，从一致性考虑，LEN 的设计上也应该同 FSPEC 保持一致，以便使用人员能够更加方便地进行数据处理。

综上所述，ASTERIX 数据格式由于缺乏统一的定义标准，各 CAT 工作组在使用过程中表现出较大的随意性，这种随意性导致了诸如“1+”和“1+1+”等类型的混淆、数据层级结构的不合理、特殊数据项使用不规范以及保留扩展字段标准不一致等一系列问题。从各类数据格式的设计初衷来看，ASTERIX 工作组似乎过于注重极致减少数据包的大小，而在规范化和可读性方面的考虑严重不足。

关于字节反转问题。在 ASTERIX 数据的解码实践中，笔者发现在接收数据时，需要对数据进行反转操作。需要进行字节反转的主要原因是：

数据项定义与内存存储的方向差异，在 ASTERIX 数据项定义中，数据包是按 CAT、LEN、FSPEC、各数据项按从低到高的顺序排列，并且每个数据项第一个字节（Octet no. 1）处于高字节位置，最后一个字节（Octet no. n）总是处于低字节位置。而在最终端处理数据时，数据在内存中的存储方向却正好相反（见图 2）。在内存存储中，字节从低位到高位分别为 CAT、LEN、FSPEC 和各数据项，而在每个数据项中，标识 Octet no. n 的第一个字节却是在低字节，最后一个字节却是在高字节。一种简单的记忆方法是：靠近 CAT 字节的数据项字节在内存中是低字节。这种 ASTERIX 数据项定义与实际存储的差异，导致在多字节情况下，必须进行字节反转操作才能保证数据的一致性。例如，对于单字节的数据项，由于不存在字节顺序的问题，所以不需要反转；对于 FSPEC 和“1+”类型，由于其存储方式是按单个字节顺序逐个储存的，在计算机内存中也是按从低位到高位排列的，是不需要进行字节反转处理的；但对于多字节的情况，如某些特定字段，就需要严格按照规则进行反转，以图 6 为

例，由于其数据定义的高字节（SAC）内存存储是低字节（更靠近 CAT），数据定义的低字节（SIC）是内存存储是高字节（远离 CAT），因此，需进行字节反转才能保证数据定义和内存存储字节顺序的一致性。

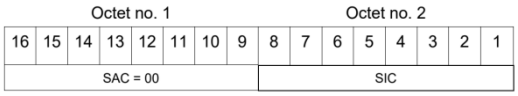

图 6 CAT001 SIC/SAC 字段

大小端模式的历史演变与冲突。在 ASTERIX 格式文档设计之初，当时的计算机主流采用大端模式，在这种模式下，数据的高字节存储在低地址，低字节存储在高地址，这与 ASTERIX 数据项定义中的字节顺序相匹配。然而，随着技术的发展，现代的计算机主流 CPU，如 AMD64、ARM、POWER PC 等，默认采用小端模式或者支持大小端反转。小端模式与大端模式相反，数据的低字节存储在低地址，高字节存储在高地址，这种历史的演变导致了 ASTERIX 早期的大小端设计与现代计算系统需求之间的矛盾，从而引发了字节反转问题。为了适应现代计算机系统，在处理多字节的数据项时，不得不进行额外的反转操作，这无疑增加了资源的消耗。

大小端问题引出其他的问题，主要有以下几种情况：（1）以图 6 所示的 CAT001 SIC/SAC 字段为例，当对类似固定长度字段内的整个数据项进行反转（称为大反转）时，会出现一些问题。比如在处理 1+a*n（此处 a=2）数据项的情况（如图 7 所示），REP 靠近 cat 字节，因此存储在低位，解码时可直接获取。但经过大反转后，REP 变到最高位，这就需要取该字段总长度的最后一个字节，增加了计算的复杂性。在实际解码中，这种方式并不方便，降低了解码效率。

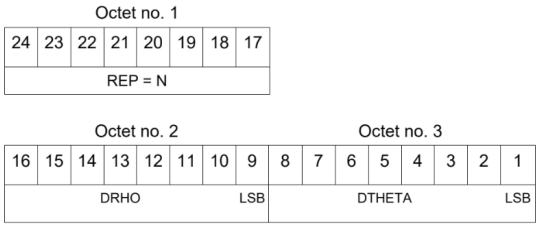

图 7 CAT010 的 PRESENCE 数据项

相比之下，不进行大反转，而是对每个重复的单个数据进行小反转是一种更优的选择。以 PRESENCE 数据为例，这样处理后，PRESENCE 高低位的存储就和定义一致了，而且每个重复项的排列顺序也与定义相符。对于 a+ 重复类型（a>1）的情况，同样适合采用这种方式。由于重复顺序是从低到高的，整体反转会破坏这种顺序，而单独对每个重复数据进行小反转则能保留原有的的排列顺序，使解码过程更加顺畅。可见，由于 ASTERIX 定义时采用大端序，计算机内存存储采用小端序，这一根本差异导致数据解码变得更加复杂。无论是大反转还是小反转，都需要额外的处理逻辑，增加了代码的复杂度和出错的可能性。同时，这些反转操作也会占用一定的计算资源，降低数据处理的速度，尤其在处理大量数据时，这种影响会更加明显。

3 ASTERIX 数据格式的改进建议

由于 ASTERIX 数据在实际应用中暴露出的问题，比如大小端序问题、保留扩展字段设计不合理等，由于历史原因和延续前后向兼容，对 ASTERIX 数据格式进行根本性改变已基本不可能，但对现有各种数据格式的重新定义和规范却十分必要，笔者强烈建议 ASTERIX 工作组进行如下修改：

3.1 数据项结构的重新定义

统一数据包基本组成为数据项：摒弃原有的五种数据项结构，将所有的数据包元素都作为数据项，建议分成五中数据项：CAT 数据项、显式长度数据项（双字节用于数据包，单字节用于保留扩展字段）、FSPEC 数据项以及基本数据项。

FSPEC 数据项的分类：将 FSPEC 分为可变长度 FSPEC（即现有的 FSPEC，可记作 FSPEC+）和固定长度 FSPEC（保留扩展字段中的 FSPEC，无 FX 扩展位）。

基本数据项的细分（1）固定长度数据项（2）a+ 类型数据项，可以适配现有的 1+，3+ 之类的数据，增强数据类型的通用性。（3）a+b*n 重复类型数据项：若 a 只能为 1，则固定为 1+b*n 类型，使重复数据的表示更加清晰。（4）x+ 类型数据项：对于类似 CAT048 的保留扩展字段 SGV 的类型，解码时可采用 [2,1] 之类的数组处理。但鉴于其应用范围较窄，更建议将该字段改为 2+ 类型，以避免新增不必要的数据类型。

3.2 定义数据块的层级结构

将《欧控规范监视数据交换 - 第 1 部分》数据块框图调整为层级结构。

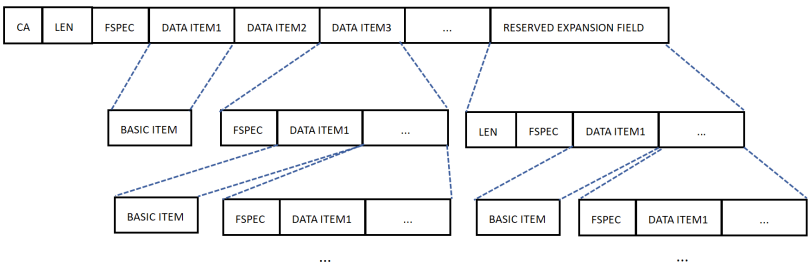

图 8 ASTERIX 层级示意图