

# 舰载情报系统模型研究

孔德宁, 杨连嘉

**摘要:** 在构建舰载情报系统过程中, 本文描述了当前海战场环境下情报处理流程的基本样式; 就一个特定领域软件 (即舰载指控系统) 进行领域分析, 设计了一个面向舰载指控领域的 CORBA 对象框架, 并应用于舰载指控系统的情报处理子系统 (情报服务器) 的开发中; 实现了该情报服务器的一个原型系统。舰载情报系统是一个实时军用信息资源系统, 本文探讨的系统构建方法亦适用于一般的信息资源系统建模与设计。

**关键词:** 舰载指控; 情报服务器; CORBA; C4ISR

## 0 引言

海军作战指挥根据情报制定作战计划, 情报是作战的依据并贯穿作战活动的全过程。现代战场作战环境日益复杂, 具有很大的不确定性, 能否取得战争的胜利很大程度上取决于获取情报的准确性、时效性和可用性。在未来信息化海战中, 战场信息不足<sup>[1]</sup>不再是困扰指挥人员的问题, “陆、海、空、天” 多级信息侦查网络提供的情报信息使得指挥人员能够实时共享战场动态、一致的态势信息<sup>[2]</sup>, 指挥人员通过 C<sup>4</sup>ISR 迅速处理复杂的战场情况, 根据变化的情况做出反应, 从而完成对各作战实体行动的有效控制。

研究舰载 C<sup>4</sup>ISR 情报模型, 旨在有效获取情报以支持作战指挥人员建立并保持对战场空间态势的综合感知, 减少战场的不确定性和未知感, 提高作战指挥决策的速度和质量。舰载情报处理软件是战术应用软件的重要组成部分, 用于分析敌方作战能力、企图、弱点和战场环境 (包括天气、气候、我方和敌方系统的天气影响矩阵、战术辅助决策能力及地形), 辅助指挥人员实施战术打击。

## 1 战术情报需求与处理流程分析

作战流程及样式大体上决定了情报任务阶段的划定。一般的, 该任务被划分为六个部分<sup>[3]</sup> (如图 1):

### a) 情报计划

制定情报计划旨在鉴别和确认情报需求, 对需求进行优先级排序, 确保情报收集机构高效运作, 从而保证情报收集、处理和分发的有序进行。制定情报计划是一个循环反复的持续过程, 经过评估后未能满足的情报需求必须被列入计划中, 以确保弥补指挥人员在战斗空间认知方面的缺失, 为目前和未来的作战提供有效支援。

情报计划与需求管理包括: 鉴别和确认情报需求; 确认情报收集、处理和分发情报需求; 对情报需求分级, 并将情报任务分发到合适层级的情报机构; 实施连续监督确保情报获取的效率。

优先级高的情报需求与战场指挥人员决策紧密相关, 影响到战场任务的完成。情报需求的优先级一般通过六个方面进行评定<sup>[4]</sup>: 能否支持指挥人员的战术判断; 能否如实反映战场态势的变化; 能否对我方战术行动提供预警; 能否为我军提供安全防护; 是否包含目标的属性及定位; 能否支持战斗评估。

### b) 情报收集

不同级别的作战指挥人员通过发送情报收集命令、情报协同命令、情报需求申请, 使各级作战系统的情报需求具体化并逐层下发, 在传感器粒度的情报搜集过程中实现信息的获取。负责收集情报资源的目标搜索系统可能是闭合回路 (武器系统内的一部分), 或开放回路 (与火力系统分离, 隶属于整个武器系统)。情报收集行动就是组织、控制与运用侦察力量在情报搜集区域内开展的搜索、探测和定位行动。情报按照收集来源划分, 可分为友方协同情报、我方搜集区域探测情报、我方上报及上级下传情报, 其中跟踪信息如距离、方位、高/深度、栅格、轨迹和速度的获取为作战系统提供简单明了的目标指示, 也为战斗毁伤评估、目标周期定位、再次攻击评估提供依据。

### c) 情报处理

情报处理包括实时接收原始探测信息、通过处理转换为可用于生成情报数据的单信源情报信息、或信号的融合处理，主要负责完成文件转换、图像生成和解释、数据解码等任务。情报处理过程中发现的情报缺失必须列入情报计划，通过再次收集得以补充。

处理和使用收集到的信息不同于情报生成，这一阶段的情报信息包含大量的数据冗余，多半不能为系统理解，必须将收集到的数据加以扬弃：将不同标准的电子数据转换为我方内部标准格式；完成图像生成和适当的解释、文件格式转换；信息相关，对相关的信息进行鉴别和编组并进行比较。这样做使得对于单个目标的数据精度提高，同时提高了信息的可靠性与可信度。

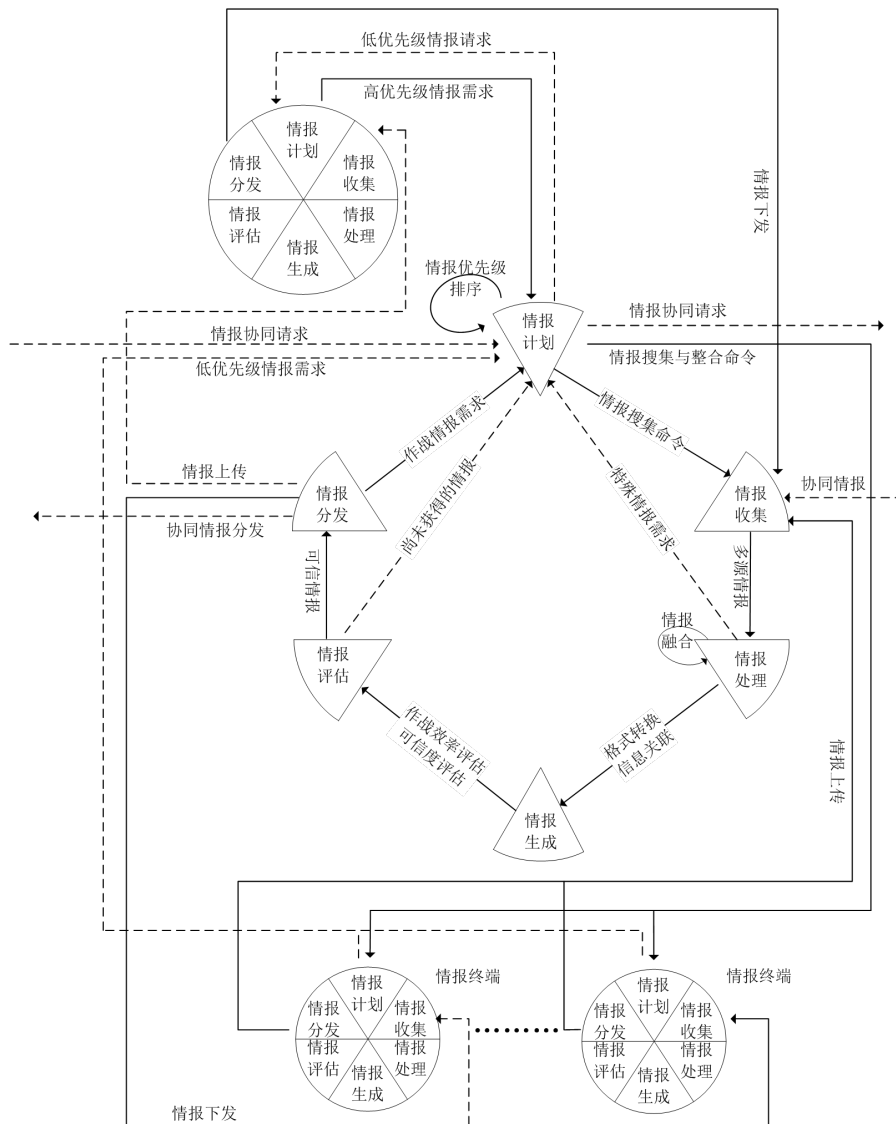


图 1 情报处理流程

a)情报生成和评估

情报生成和评估是汇集多情报源提交的、经过情报处理可为舰载自动化作战系统识别的情报信息，进行评估、集成、分析和综合，并将其解释为最终的情报数据的过程。通过集成、解释、分析和综合，收到的信息将与已有的情报共同转换为适合使用、关乎任务的情报，从而满足指挥人员的情报需求。信息分析和综合是生成情报的重要步骤，主要为战场指挥人员形成威胁确认、对战斗环境的详细说明、对战斗空间影响评估、威胁评估、判断敌方行动方案。

评估信息包括评价情报的相关度、可靠性和准确度。评估信息表明信息与实际情况的相关度、可靠性和精确度，将数据、信息及态势进行集成则有助于形成敌方能力、局限以及我方行动方案合乎逻辑的评估。

利用该评估得出当前问题和主要威胁是生成情报的意义所在，这样的评估包含对战场环境（包括战场地区的政治、经济、工业、地理、人口、地形、水文、气候，以及人文因素，如人口文化、教育、医疗、语言、历史和心理特征）详细说明。情报评估将未满足的情报需求汇总到情报计划中，通过情报收集满足缺失。

### b)情报分发

情报分发选择何种形式（音频、文字、图形格式诸如图表、图像、透明图等、标准单一来源或多源的情报报告、简报格式或电子格式），取决于情报是否能够最大限度满足指挥人员的实时需要且使用简便。情报分发要求安全快速，通过数据链传输是情报分发的重要途径。

情报分发单元的核心是信息分发管理服务，它与情报处理单元联动，按照情报计划机构制定的分发策略，对经过评估的情报产品进行集中分发，并由对应的网络传输服务将已编码的情报数据发送最终用户。

## 2 舰载情报模型

舰载战术情报模型的建立基于情报信息在作战系统的流动方式，也基于作战部队的组织体系结构。参与区域多军、兵种联合作战活动的海军情报节点可能是航母编队或海上战斗舰船群中的任意某舰，即可通过目力、听力、电子探测、照相等所有手段采集情报并可与编队中其他舰船实时共享图片、音频或标准格式报文的机动作战单舰。

从单舰的视角出发，舰船编队所在的整体作战空间是涉及联合参与该战术级作战活动的所有作战单元关注区域的总和。某一海军作战单元关注区域在实际作战活动中不仅仅是其独立作战区域，还应包括该作战单元机动位置对应的作战区域，其协同作战的作战区域，以及与当前潜在的敌方威胁相触及的区域。

从联合编队的视角出发，多种传感器同时进行目标搜索，不同的传感器都向指控系统传回自己的搜索结果信息，因此指控系统需要对所有的信息进行数据融合，通过节点情报服务器综合分析判断，做出一个统一的战场态势呈现给战场指挥员。

情报服务器是舰载作战系统的中枢，主要功能包括：接受外部设备发送的目标信息；根据融合算法对目标进行融合计算，输出统一的战场目标态势；根据用户界面命令对目标进行各类操作，增加、删除、锁定、上报等；提供情报查询等相关服务。情报服务器相当于指控系统的情报处理中心，是指控系统信息处理的核心部分，也是指控系统的一个基础系统，指控系统随后的信息处理（包括设备分控制、武器系统的指挥等）都是建立在情报服务器输出的战场统一态势的基础之上的。

### 2.1 情报服务器

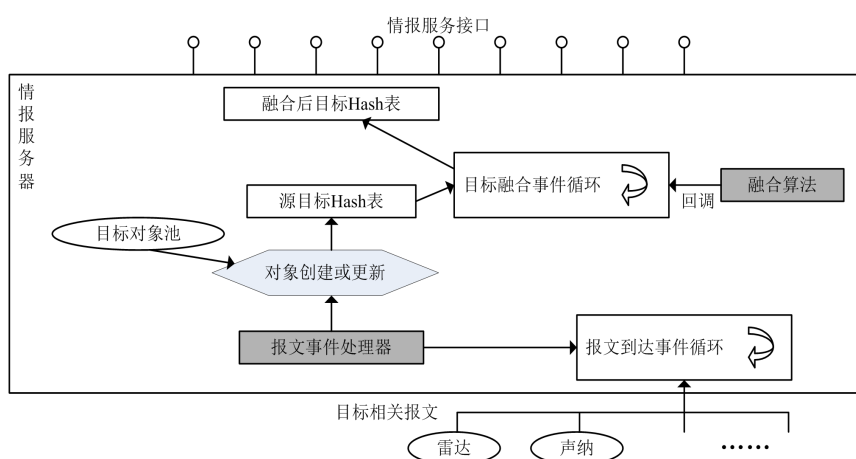


图2 情报服务器应用实例

要建立情报服务器的体系结构，首先应区分指控系统中比较稳定的部分（即系统共性）和不稳定的、随不同项目而变化的部分（即系统特性）。对情报服务器而言，共性就是抽象目标的完整处理流程，个性就是各种不同格式的报文和目标融合的算法。图2描述了舰载指控系统的情报处理的一个应用实例。

情报服务器分离出了情报处理过程中的不变和可变部分：报文事件处理器和目标融合算法是系统中最易发生变化的，而两个异步事件循环流程（报文到达事件和目标融合事件）基本上是稳定的。对于报文的可变性，情报服务器通过事件循环回调事件处理器的方式把不同的报文处理纳入框架；对于融合算法，目标服务框架定义了目标融合算法的 CORBA 组件接口，只要不同的融合算法实现遵从该接口，目标服务框架就很容易替换不同的算法。

目标融合事件是一个定时事件，它根据融合算法对从报文事件循环创建的目标对象进行数据融合，得出战场的统一态势，融合后的对象保存在融合后目标 Hash 表中。融合流程如图 3。

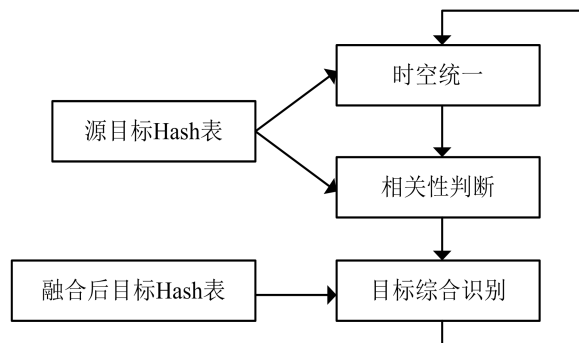


图 3 情报服务器数据融合流程

情报服务器还定义了两个全局的 Hash 表（源目标 Hash 表和融合后 Hash 表），源目标 Hash 表用于保存最初的目标航迹对象（由目标报文解析而来），而融合后的 Hash 表用于保存融合后的统一目标航迹表。框架中很多操作都涉及到对这两张表的读取和改写，因此对这两张表的操作是整个系统性能的关键。

目标对象池是存放预定义目标对象的容器，它利用空闲时间保证池内存在一些预定义目标对象，当报文事件处理器需要创建目标时就可直接从池里获取，有助于提升系统处理并发数据的性能。

## 2.2 情报服务器的实现

情报服务器的实现首先要明确它必须管理的对象和事件。舰载指控系统情报处理最关注的就是目标航迹对象、传感器对象、武器对象等实体对象。由舰载指控系统自身的性质所决定，这三类对象所含的信息量（对象属性）比较多，并且不同的信息存在较大的差异，因此有必要把这些对象按照继承和组合的方式进行合理的划分。情报服务器的事件包括了非周期性的报文输入事件和周期性的目标融合事件。报文输入事件，主要是各类报文解析器的编写，并在事件循环注册后由事件循环自动回调；目标融合事件，主要是融合算法接口的制定，它必须能确保融合算法的可变性。下面本文将详述情报服务器类的设计。

### 2.2.1 类的设计原则

利用面向对象的设计方法，可在情报服务器体系结构下详细设计类的结构与关联。为了增强系统的健壮性、适应性、灵活性、可维护性、可重用性，并降低系统内部各组件之间的耦合度，本文在设计情报服务器系统类结构与关联时应用了下列设计原则<sup>[5-9]</sup>，并贯穿详细设计全过程。

#### a) 开放/封闭性原则（The Open Closed Principle, OCP）。

该原则要求：对于扩展，模块必须是开放的；而对于修改，模块必须是封闭的。适当的抽象是 OCP 原则的关键。

#### b) Liskov 替换原则（The Liskov Substitution Principle, LSP）。

LSP 原则最早是 Barbar Liskov 在数据抽象与类型理论的研究中提出来的。该原则要求：派生的子类应该能替代基类。如图 4 所示，当用派生的子类传递给一个基类的用户时，该用户应该还能继续正常的工作。

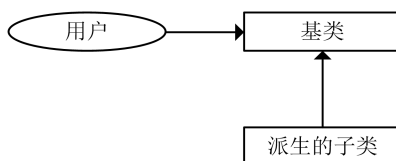


图 4 LSP 原则

c)依赖性倒转原则 (The Dependency Inversion Principle, DIP)。

DIP 原则要求：依赖于抽象类，而不依赖于具体的类。本文在详细设计情报服务器系统时，应用了“抽象工厂”等设计模式以满足该原则的要求。

d)接口隔离原则 (The Interface Separation Principle, ISP)。

ISP 原则指明：多个特定于用户的接口比一个通用性目的的接口更好。例如，如果一个服务器提供给多个用户不同的接口，则应当对服务器所提供的通用性接口进行封装与隔离。

### 2.2.2 类的设计

本文设计情报服务器类图如图 5 所示，所有的类按属性可分为三类：实体类、控制类和接口类。实体类包括对原指控系统情报处理部分航迹信息结构的封装。由于该结构比较庞大，因此定义了一个基类 TargetRoot，并由此派生出三个子类：航迹目标类 TrackTarget、平台目标类 PlatformTarget 和电子战目标类 EWTarget。控制类相当于对象管理器的对象管理接口，负责实体类对象的创建、查询和删除操作。接口类相当于事件管理器的事件处理器，包括了报文解析器类 MsgHandle 和融合算法类 FusionAlgorithm。

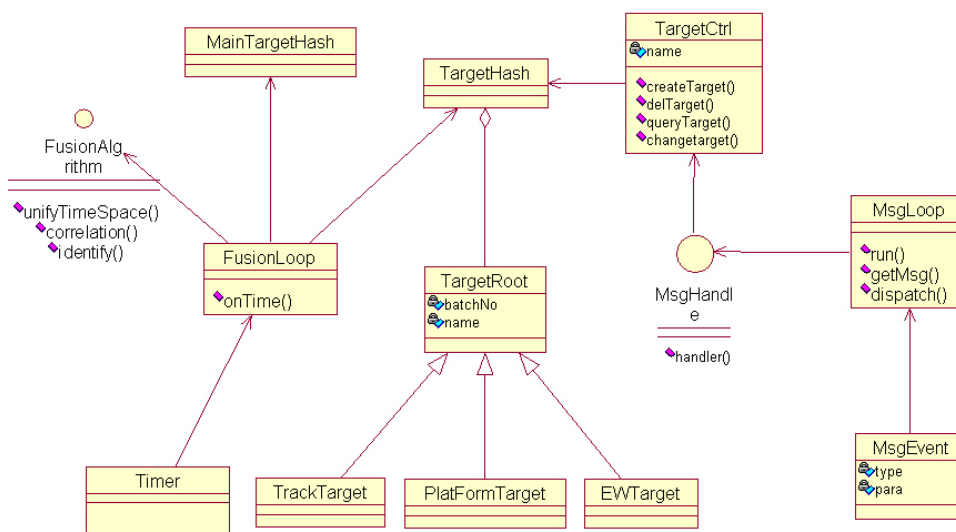


图 5 情报服务器类图

情报服务器中包含两个主要的处理流程：一是报文解析处理流程，其顺序图如图 6 所示；二是数据融合流程，其顺序图如图 7 所示。

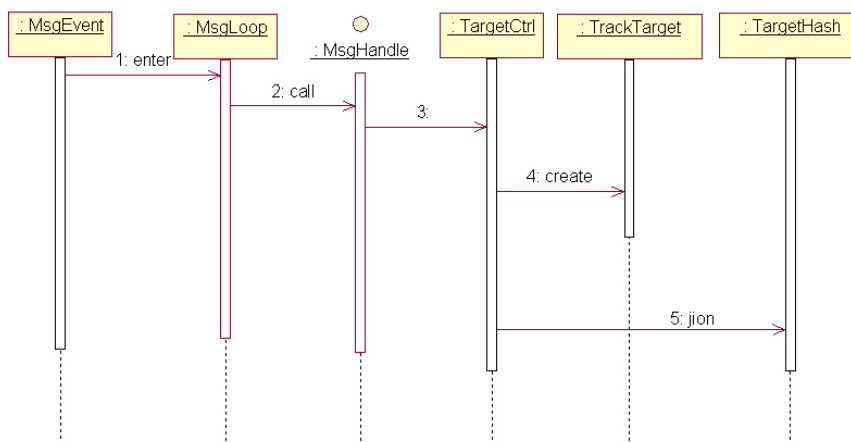


图 6 报文解析处理流程

图 6 的报文处理流程中，首先由情报数据的到达引起报文数据到达事件。该事件被调度到报文处理循环中，报文处理循

环根据事件类型回调相应的报文解析器，接着报文解析器调用对象管理器的接口函数创建或更新目标对象，最后创建的目标对象被加入目标对象 Hash 表中。这样就完成了一次报文处理的全过程。

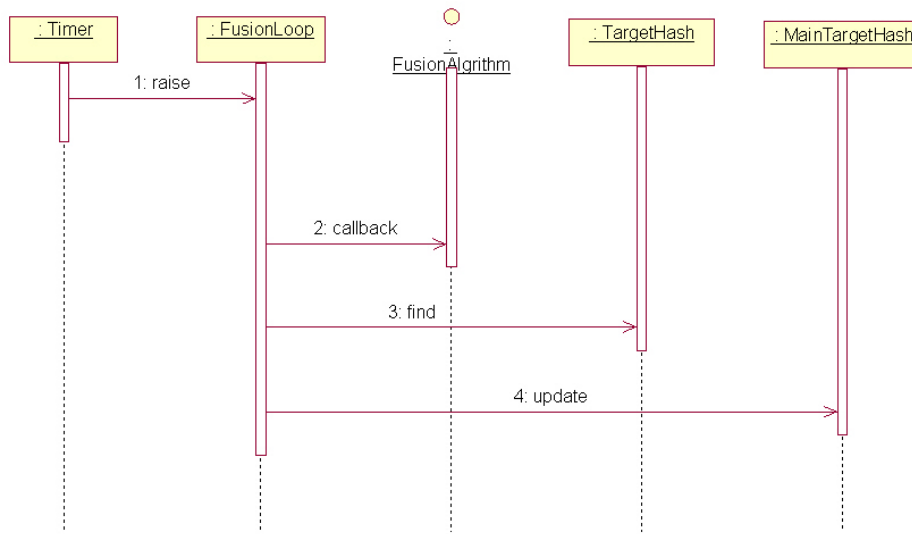


图7 目标融合流程

图7的目标融合处理流程中，首先由定时器引发定期融合事件，融合循环过程开始。融合过程遍历目标 Hash 表，并调用融合算法对目标进行融合判断，将融合后的结果保存入主航迹 Hash 表内。当遍历完所有的目标对象后，一次融合过程就结束了。

### 3 结束语

本文描述的情报服务器是指控系统情报处理的一个软件重用解决方案，通过对原有的情报信息处理进行重构与对象封装，对外以中间件接口的形式对上层应用提供目标相关服务。情报服务器有自己的控制流和可变点的行为调整机制，最大程度的抽象出指控系统中情报处理的共性特征。经过近年来的研究与实践，对象框架技术已在公共服务领域获得了较大发展，但在特定领域，特别是军用系统软件领域的研究还有待于进一步深入，本文为此作了有益的探讨和尝试。

#### 参考文献：

- [1]克劳塞维茨.战争论.译林出版社.2011年.
- [2]孔德宁,杨连嘉.基于CORBA的雷达组网领域服务的设计与实现.国防科技报告,2009.
- [3]孔德宁,杨连嘉.海军作战战术分析.国防科技报告,2012.
- [4]Navy Tactical Task List(3.0).2007.
- [5]Thomas C.Jepsen,Joseph Williams.Software in the New Millennium:A Virtual Roundtable. Software Development,IT Pro.2010,8.
- [6]Douglas C Schmidt,Aniruddha Gokhale,Balachandran Natarajan.Leveraging Application Frameworks.Queue of ACM.2009,2(5).
- [7]Leesa Murray,David Carrington,Paul Strooper.An approach to specifying software frameworks. Proceedings of the 27th conference on Australasian computer science.2004,26:185-192.
- [8]Jan Bosch,Peter Molin,Michael Mattsson,PerOlof Bengtsson.Object-oriented framework-based software development:problems and experiences.ACM Computing Surveys.2010,32(1).
- [9]Damien Touraine,Patrick Bourdot,Yacine Bellik,Laurence Bolot.A framework to manage multimodal fusion of events for advanced interactions within virtual environments. ACM International Conference Proceeding Series Proceedings of the workshop on Virtual environments. 2002, Pages:159-168.