

# 一种信息汇聚服务器套件的设计与实现<sup>①</sup>

邵东青, 李 峰, 孙 义, 刘冰峰

(63892 部队, 洛阳 471003)

**摘 要:** 为解决军事训练中的装备活动信息汇聚问题, 参照即时通信系统设计实现一种以信息汇聚服务器为中心的软件套件。该套件采用内容可定制的信息模型以及长度可变的信息报文, 实现信息汇聚、信息下载等业务功能以及基于 Web 的服务器监控功能, 能够一致处理不同类型的装备活动信息, 允许通过 Web 对汇聚服务器进行监控, 满足了相关训练信息系统的信息汇聚需求。

**关键词:** 信息汇聚; Windows 服务; 服务器监控; Agent; XML

## Design and Implementation of an Information Gathering Server Suite

SHAO Dong-Qing, LI Feng, SUN Yi, LIU Bing-Feng

(Unit 63892 of PLA, Luoyang 471003, China)

**Abstract:** To solve the problem about gathering equipment activity information in the military drilling, a software suite, in which the information gathering server is central, is designed and implemented, referring to the instant message systems. The suite adopts a content customizable information model and a set of length variable information messages, implements functions such as information gathering, information downloading, and web-based server monitoring. It can uniformly handle various types of equipment activity information, allows monitoring the gathering server through Web, and meets the requirement for information gathering of concerned drilling information systems.

**Key words:** information gathering; Windows service; server monitoring; agent; XML

军事训练中, 参训装备会开展一系列的活动。为了及时感知参训装备的状态以及训练场区的对抗态势, 训练组织方需要实时采集并上报各类参训装备的活动信息。由于大规模训练中会使用为数众多的采集终端, 同时会存在超过一个的应用系统, 如何汇聚终端设备所采集的装备活动信息, 并将其转发给最终应用系统, 已成为军事训练信息化中急需解决的关键问题之一。由于装备活动信息的汇聚是与装备活动同步进行的, 传统以信息管理为目的软件已不再适用, 为此, 某项目中参照即时通信系统<sup>[1-4]</sup>, 设计实现了一种信息汇聚服务器套件, 本文将对其总体设计及其关键功能的实现进行探讨。

## 1 总体设计

### 1.1 总体架构设计

为了满足各类装备活动信息的汇聚需求, 同时支持外部集成与监控, 该套件被设计成一组以汇聚服务器为中心、结构相对独立、功能相互配合的软件部件及 Agent 类库, 其总体架构如图 1 所示。

其中:

① 汇聚服务器: 是套件的核心部件, 以 Windows 服务形式实现, 运行在后台服务器上。它提供信息汇聚服务, 接收并转发信息; 也提供信息下载服务, 允许下载其中保存的信息; 另外还提供对自身的监控服务, 允许远程管理其所提供的业务服务。

<sup>①</sup> 收稿时间:2011-10-19;收到修改稿时间:2012-01-10

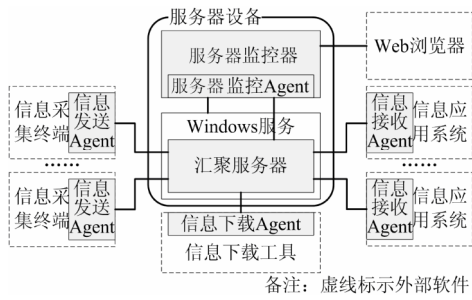


图 1 套件总体架构

② 信息发送 Agent: 以类库形式实现, 封装采集终端与服务器的交互细节, 通过 TCP/IP 协议同汇聚服务器交互, 为信息发送提供访问接口。

③ 信息接收 Agent: 以类库形式实现, 封装应用系统与服务器的交互细节, 通过 TCP/IP 协议同汇聚服务器交互, 为信息接收提供访问接口。

④ 信息下载 Agent: 以类库形式实现, 封装下载工具与服务器的交互细节, 通过 TCP/IP 协议同汇聚服务器交互, 为信息下载提供访问接口。

⑤ 服务器监控 Agent: 以类库形式实现, 封装监控器与汇聚服务器的交互细节, 通过 Windows 服务控制器同服务器的宿主服务交互, 通过命名管道同服务器的业务服务交互, 为服务器监控提供访问接口。

⑥ 服务器监控器: 是 ASP.NET 应用, 用于监控汇聚服务器, 部署在汇聚服务器所在的服务器上。它通过服务器监控 Agent 同汇聚服务器交互, 允许用户通过 Web 以 B/S 模式远程监控汇聚服务器。

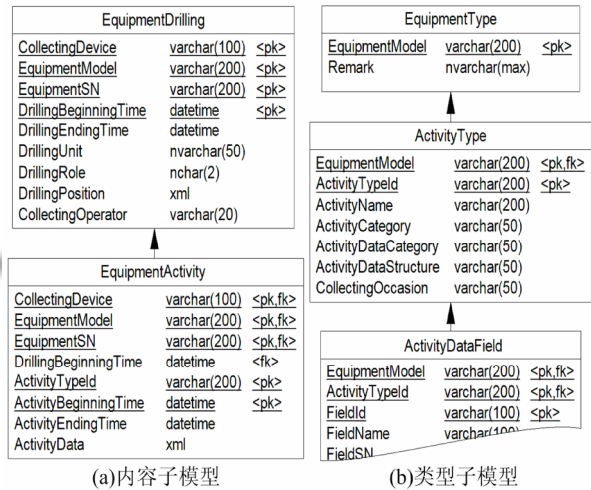
### 1.2 信息模型设计

该套件需要支持各型装备, 而不同装备的活动信息的组成不尽相同、无法预知, 传统的静态信息模型设计已无法满足该套件中信息处理与管理的需要。为此, 该套件中采用了如图 2 所示的内容可定制的信息模型, 它在正常的内容子模型之外还包含了用于内容定制的类型子模型。

其中:

① 类型子模型: 描述装备活动信息的类型, 类似于关系数据库中的数据字典, 包括表 EquipmentType、ActivityType 和 ActivityDataField。其中, 表 EquipmentType 描述装备的类型信息, 用于分组管理装备活动信息的类型信息; 表 ActivityType 描述装备活动信息的总体类型信息, 它利用字段 ActivityCategory 将装备活动划分为过程和动作等两大类, 同时规定了装备活动

数据的类别及结构; 表 ActivityDataField 描述装备活动数据的组成字段信息, 包括字段的标识、名称、顺序、类型等信息。这个子模型允许用户定制装备活动信息, 而不用维护套件实现, 其所描述的定制信息可事先分配给采集终端和应用系统以便它们生成和解析具体的装备活动信息。



(a)内容子模型 (b)类型子模型

图 2 信息模型组成

② 内容子模型: 描述装备活动信息及其相关训练信息, 包括表 EquipmentDrilling 和 EquipmentActivity。其中, 表 EquipmentDrilling 描述装备训练信息, 说明装备活动的相关情境; 表 EquipmentActivity 描述装备活动信息, 它以 XML 字段 ActivityData 统一保存各类装备活动数据, 而不关注其具体组成, 这避免了传统关系模型中需要针对不同类型不断建表的困境。特别指出, 仅依据表 EquipmentActivity 的定义并不能完全理解其中的内容, 如果需要, 则还应通过其中的字段 EquipmentModel 和 ActivityTypeId 引用对应的类型信息, 正是由于这种设计, 该套件实现装备活动信息一致处理的同时确保了其具体活动数据的可解析性。

### 1.3 信息报文设计

信息汇聚时需要使用信息报文传送图 2(a)所示的信息。依据上述信息模型, 并根据活动本身的时间特性以及信息采集的时机要求, 该套件使用 6 种报文即可完成所有信息的传送。其中, 训练信息使用训练开始和训练结束报文; 过程类活动信息可能使用过程开始和过程结束报文, 也可能使用过程报文; 而动作类活动信息则只使用动作报文。由于活动数据长度因类型不同而不同, 为了确保套件适应性, 不能采用长度

固定的报文格式。因此，为了规范和简化相关模块的处理，该套件中采用了基于 XML 的可变长报文格式，同时对有关报文进行了如图 3 所示的封装，由它们处理报文对象到 XML 字符串再到字节流的序列化以及字节流到 XML 字符串再到报文对象的反序列化。

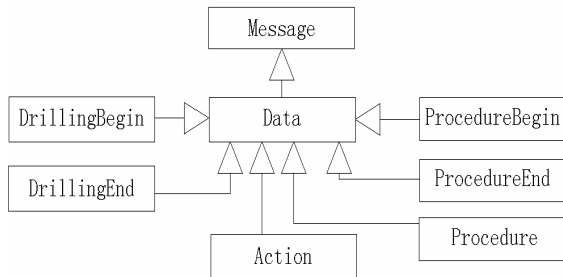


图 3 信息报文组成

其中，Message 类是抽象类，实现 XML 格式的对象字符串到字节流的转换，信息报文以及其它网络报文都由其派生；Data 类也是抽象类，封装所有信息报文共有属性，具体信息报文由其派生；DrillingBegin、DrillingEnd、ProcedureBegin、ProcedureEnd、Procedure 和 Action 类分别封装训练开始、训练结束、过程开始、过程结束、过程和动作等信息，它们利用 XML 来处理可变长报文的序列化/反序列化，具体做法是：在报文对象被序列化成字节流前，先使用 W3C 文档对象模型(DOM)将其序列化成 XML 字符串；在从字节流中提取出对象字符串后，使用 XPath 数据模型<sup>[5]</sup>解析其并生成相应的对象。

## 2 关键功能实现

### 2.1 信息汇聚

信息汇聚是该套件所要支持的一项核心功能，它由汇聚服务器与信息发送、接收 Agent 协作实现，它们一起构建从采集终端到应用系统的信息传送链路，其相关架构与通信图如图 4 所示。

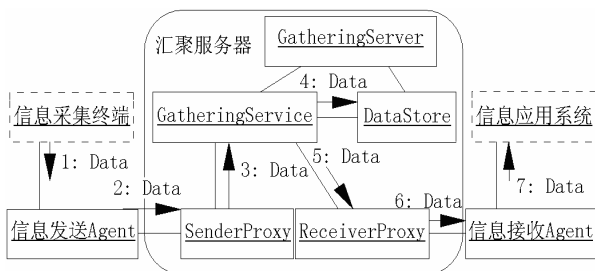


图 4 信息汇聚架构与通信图

为此，汇聚服务器内部设计实现了信息汇聚服务，其相关架构，除了容器类 GatheringServer 以及处理数据库交互的数据存储类 DataStore 外，还涉及以下多个对象类：

① SenderProxy 类：信息发送客户的 Proxy，处理与信息发送客户的交互细节。它接收信息报文字节流，将其转换为报文对象后提交给上层服务。

② ReceiverProxy 类：信息接收客户的 Proxy，处理与信息接收客户的交互细节。它序列化信息报文对象，将其通过网络发送给应用系统。

③ GatheringService 类：封装信息汇聚逻辑，为登录的采集终端和应用系统创建相应的 Proxy 实例，经由它们收发信息报文，如果需要，也可将信息报文提交给数据存储。

另外，关于信息汇聚通信，如图 4 所示它采用了推送模式。为了提高信息推送的健壮性，相关模块的实现中应用了一些专门的设计。

#### ① 通过队列和信号实现信息的异步推送

信息推送时，需要通过网络传输以及向数据库写入，这些都是费时操作。特别地，对于网络传输，可能因为链路原因而不能完成。如果采用同步推送的方式，则可能由于一个环节的阻塞而引起整个链路的阻塞，严重时会影响到外部系统的运行。为此，在数据发送 Agent、ReceiverProxy、数据接收 Agent 以及 DataStore 中增加了专门的缓冲队列，信息推送时只需将信息放入其中，并设置相应的“信息可得到”信号即可返回，而它们的内部线程则会根据信号指示循环检查缓冲队列，完成实际的推送操作。

#### ② 缓存开始信息以确保信息接收的完整性

如 1.3 节所述，信息汇聚时，有些信息需要分两次发送，如果不保存开始信息，则当应用系统在开始信息推送完毕后才登录，就收不到完整的信息，这会造成信息接收的混乱。为此，在 SenderProxy 中增加了开始信息缓存列表，它的实例接收到开始信息后，在正常推送的同时还在该列表中进行保存，当有新的应用系统登录时，信息汇聚服务将检查所有的 SenderProxy 实例，将它们缓存的开始信息一次性地推送给当前登录的应用系统。

### 2.2 信息下载

为管理信息汇聚过程中服务器上保存的信息，该套件还需提供信息下载功能，它由汇聚服务器和信息下载

Agent 协作实现，其相关架构与通信图如图 5 所示。

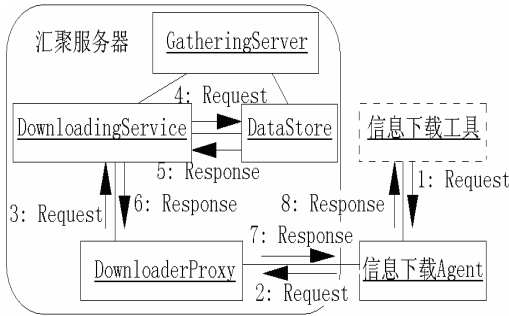


图 5 信息下载架构与通信图

为此，汇聚服务器内部设计实现了信息下载服务，其相关架构，除了信息汇聚提及的类 GatheringServer、DataStore 外，还涉及以下两个对象类：

① DownloaderProxy 类：信息下载客户的 Proxy，处理与信息下载客户的交互细节。它接收并提交来自信息下载工具的请求，接收并发送来自信息下载服务的响应。

② DownloadingService 类：封装信息下载逻辑，为登录的信息下载工具创建相应的 Proxy 实例，并通过 DataStore 执行具体的信息管理工作。

另外，关于信息下载通信，如图 5 所示它采用了请求/响应模式。与信息报文一样，信息下载中的交互报文也是通过对象类封装的，其中装备活动信息及其相关训练信息以 XML 文档形式插入到响应报文对象中。为了支持该 XML 文档的处理，该项目中使用了由图 2(a)所示子模型生成的数据集，通过它实现信息对象与 XML 文档之间的转换<sup>[6]</sup>。

### 2.3 服务器监控

汇聚服务器运行在后台服务器上，为方便管理，该套件还需提供服务器远程监控功能，它由汇聚服务器、服务器监控 Agent 和服务器监控器协作实现，其相关架构与通信图如图 6 所示。

如 1.1 节所述，服务器监控实际上包括两方面：一是宿主服务监控，二是业务服务监控。为此，汇聚服务器内部设计实现了业务服务监控服务，外部封装成符合标准的 Windows 服务，其相关架构，除了类 GatheringServer、GatheringService、DownloadingService 外，还涉及以下两个对象类：

① ServerHostService 类：是汇聚服务器的宿主服务类，从 Windows 服务基础类派生，可通过 Windows

服务控制器对其实例进行管理，当它的实例被启动或关闭时，相应地启动或关闭汇聚服务器。

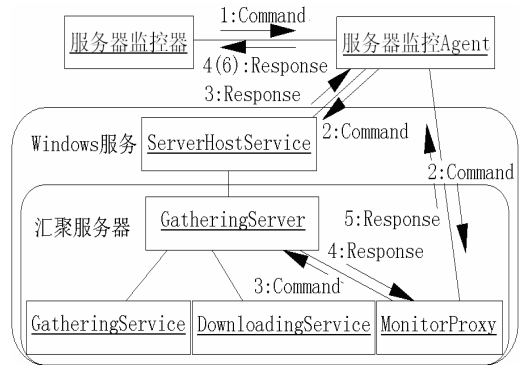


图 6 服务器监控架构与通信图

② MonitorProxy 类：以单体模式<sup>[7]</sup>实现，封装业务服务监控服务，它通过命名管道接收监控命令和发送处理结果，通过上层的 GatheringServer 实现对信息汇聚服务和信息下载服务的监控。

另外，关于服务器监控器，如 1.1 节所述，它是 ASP.NET 应用，它给用户的监控操作带来方便，但也有网络流量大、反应时间长等不足，以及多人并发操作而带来监控互扰的问题。为此，该项目中除了启用 Ajax<sup>[8]</sup>技术支持异步和部分页面更新以改善用户的操作体验外，为了避免监控操作互扰问题，特地采用了监控 Agent 全局唯一、共享的方案，这保证了所有监控都通过同一个 Agent 完成，再加之监控 Agent 本身的线程安全性设计，从而保证了监控逻辑不会因为多人同时访问而被破坏。

### 3 结语

为解决军事训练中装备活动信息的汇聚问题，某项目中设计实现了一种信息汇聚服务器套件，本文对其总体设计及其关键功能的实现进行了探讨。该套件已成功地应用于有关训练信息系统的建设中，满足了它们对装备活动信息的汇聚需求。由于采用了内容可定制的信息模型以及长度可变的信息报文，该套件能够以一致的方式处理各类装备活动信息，较好地克服了具体训练中参训装备和活动数据复杂多变而带来困扰。另外，由于采用了基于 Web 的管理方式，使得用户可以通过 Web 远程监控汇聚服务器，这为汇聚活动本身的管理提供了便利。

(下转第 190 页)

数据库方面的知识,学习 ASP 语言以前需要学习 Vbscript 及 Jscript 方面的知识,同时需要通过专业外语翻译相应的专业术语。

从模拟实验可以看出, B-P 认知模型是有效的和可行的,该模型是一个统计分析模型,随着样本信息的更新,此模型的结果会呈动态变化。

与其他的认知模型相比, B-P 认知模型可以从以往学习者杂乱的学习路径中按当前学习者的学习要求挖掘出最适合学习者的学习轨迹,这样的学习轨迹可以剔除以往学习者无效的学习步骤,极大地提高学习者的学习效率。 B-P 认知模型的不足之处是模型生成效果受以往学习者信息量大小的制约,当以往学习者信息量较小时其模型生成效果较差。

## 5 总结

本文针对目前学习者从固定认知结构获得认知模型的局限性,就认知元素固定而认知模型不固定的学习行为,借助贝叶斯推理和 PFNET 理论提出了学习者通过统计其他学习者学习轨迹的样本信息生成自己需要的 B-P 认知模型,以指导自己的学习。本文对该

认知模型的研究尚处在初级阶段,相应的实验平台未搭建成功,本人在今后将继续这方面的工作。本文的研究具有一定的特色,其研究对同类其他研究具有一定的借鉴指导意义。

## 参考文献

- 1 王新鹏.认知模型研究综述.计算机工程与设计,2007,28(16):4009-4011.
- 2 李锐,凌云翔,老松杨.基于双手交互的指挥空间响应模型.计算机工程,2008,7,34(14):274-276.
- 3 张淑军,班晓娟,陈勇,等.基于记忆的人工鱼认知模型.计算机工程,2007,33(19):33-35,38.
- 4 百度百科.[2011-07-10].http://baike.baidu.com/view/1797692.htm.
- 5 Schvaneveldt RW. Pathfinder Associative Networks Studies in Knowledge Organization. Norwood: ABLEX Publishing Corporation. 1989.1-20.
- 6 Ricardo BY, Berthier RN.王知津,贾福新,郑红军译.现代信息检索.北京:机械工业出版社,2004.34-35.

(上接第 134 页)

## 参考文献

- 1 于少山,卡米力,毛依丁.基于 XML 的即时通信系统的研究与实现.重庆邮电大学学报,2007,(6):59-61.
- 2 郭阳勇.企业即时通信软件的设计与实现.电脑编程技巧与维护,2010,(24):85-86.
- 3 陈立浩.基于 B/S 和 C/S 的即时通信系统.计算机工程,2009,35(15):270-271.
- 4 宁国强,李谢华,尹张飞.一种安全的即时通信解决方案.计算机工程与应用,2007,47(3):82-83.
- 5 莫佳.XPath 在 XML 查询中的应用.重庆三峡学院学报,2008,24(3):51-53.
- 6 Nagel C, Evjen B, Glynn J. C#高级编程.第 6 版.北京:清华大学出版社,2008.854-862.
- 7 刘必广.C#私有构造函数在设计模式中的应用.长春师范学院学报(自然科学版),2009,28(3):66-69.
- 8 Evjen B, Hanselman S, Rader D. ASP.NET 3.5 高级编程.第 5 版.北京:清华大学出版社,2008.849-881.