基于警务业务的低代码组件化颗粒度分析

张铁龙[1] 姚叙辰[1] 李巧霞[2] 赵全举[2] 杜林佳[2]
1. 河北雄安新区公安局，河北 雄安 071000
2. 北京安融畅信息技术有限公司，北京 102300

[摘要]文中旨在探讨基于警务业务的低代码组件化颗粒度分析。随着信息技术的不断发展，警务工作也在积极寻求创新与变革。低代码开发平台的出现为警务业务的信息化建设提供了新的途径，而组件化设计则有助于提高系统的可维护性和可扩展性。文中将深入剖析警务业务的特性，综合考虑多种因素，构建科学合理的低代码组件化颗粒度分析模型，并通过警情处置案例验证其有效性提高警务工作效率。

[关键词]警务业务；低代码；组件化；颗粒度分析
DOI：10.33142/sca.v8i1.15090 中图分类号：TP311.5 文献标识码：A

Particle Size Analysis of Low Code Componentization Based on Police Business

ZHANG Tielong[1], YAO Xuchen[1], LI Qiaoxia[2], ZHAO Quanju[2], DU Linjia[2]
1. Public Security Bureau of Hebei Xiong'an New Area, Xiong'an, Hebei, 071000, China
2. Beijing Anrongchang Information Technology Co., Ltd., Beijing, 102300, China

Abstract: This article aims to explore low code component-based granular analysis based on police operations. With the continuous development of information technology, police work is also actively seeking innovation and change. The emergence of low code development platforms provides a new approach for the informatization construction of police operations, while component-based design helps to improve the maintainability and scalability of the system. In this article, we will deeply analyze the characteristics of police operations, comprehensively consider various factors, construct a scientific and reasonable low code component-based granular analysis model, and verify its effectiveness through police case handling to improve police work efficiency.

Keywords: police business; low code; componentization; particle size analysis

引言

低代码开发平台因快速开发、易维护等优点在警务领域渐获应用，同时引入的组件化设计可分解复杂警务业务系统为独立组件。低代码组件化应用于警务业务时，核心在于确定组件化颗粒度，即界定组件大小、功能范围与抽象层次等，其合适与否对打造优质、易维护且可灵活扩展的警务低代码应用系统起决定性作用。组件化颗粒度若太粗糙，组件功能庞杂、逻辑难懂难维护，复用性差，业务有变时也难针对性调整优化；而颗粒度太精细，会提升组件交互复杂度，催生系统性能瓶颈，还让开发人员在选组、组合时决策更纠结，致开发效率降低。因此，对基于警务业务的低代码组件化颗粒度进行深入分析与研究，探索适合警务业务特点的最佳组件化颗粒度策略，对于充分发挥低代码开发在警务信息化建设中的优势，提升警务业务应用系统的整体质量和适应性具有极为关键的理论与实践意义。

1 低代码开发在警务业务中的应用

1.1 低代码的定义与特点

低代码开发是一种通过可视化界面和少量代码来快速构建应用程序的开发方式。它具有模块化、可视化以及灵活开发等特点。在警务业务中，低代码开发可以降低开发成本，满足不断变化的业务需求，全面提升开发效率。

1.2 警务业务中的低代码应用场景

警务业务里，低代码应用丰富。在案件管理中，接警到结案流程借助它高效流转、文书规范生成；交通执法支持现场快速处理、线上操作；应急指挥一键调配资源；装备管理扫码出入库，精准预警。

2 组件化设计在警务业务中的重要性

组件化是将一个复杂的系统分解为多个独立的组件，每个组件都具有明确的功能和接口。在警务业务中，组件化设计具有以下优势：

2.1 提高系统的可维护性

当系统出现问题时，快速定位到具体的组件进行修复，而不会影响整个系统的运行。

2.2 增强系统的可扩展性

可以根据业务需求的变化，方便地添加或替换组件，实现系统的功能扩展。

2.3 提高开发效率

开发人员可以复用已有的组件，减少重复开发的工作量，提高开发效率。

Copyright © 2025 by authors and Viser Technology Pte. Ltd.

3 颗粒度分析在低代码组件化中的作用

3.1 颗粒度的概念与分类

颗粒度是指组件的大小和功能的细化程度。颗粒度可以分为粗颗粒度、中颗粒度和细颗粒度。粗颗粒度的组件功能较为强大，但灵活性较差；细颗粒度的组件功能相对单一，但灵活性较高。

3.2 颗粒度对低代码组件化的影响

3.2.1 对系统性能的影响

颗粒度的大小会影响系统的性能。细颗粒度的组件可以提高系统的灵活性和可扩展性，但会增加系统的复杂度和开发成本。粗颗粒度的组件可以降低系统的复杂度和开发成本，但会降低系统的灵活性和可扩展性。

3.2.2 对开发效率的影响

合理的颗粒度划分可以提高开发效率。如果颗粒度过细，开发人员需要花费大量的时间进行组件的组合和调试；如果颗粒度过粗，开发人员可能需要进行大量的重复开发工作。

4 构建警务业务低代码组件化颗粒度分析模型

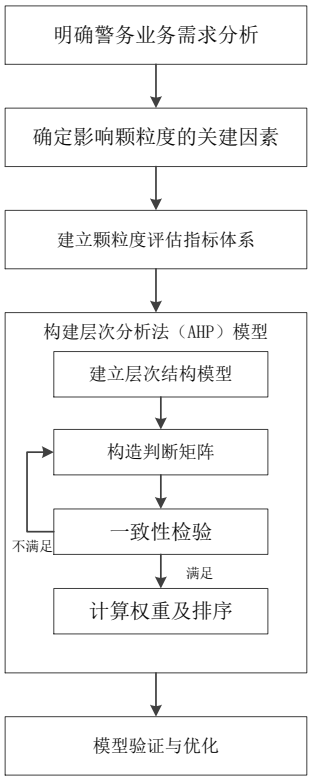

图 1 构建警务业务的低代码组件化颗粒度分析模型流程图

4.1 明确业务需求和目标

4.1.1 梳理警务业务流程

以治安管理为例，明确业务流程。

治安管理流程：包括日常巡逻任务分配、巡逻路线规划、治安事件接警、现场处置、案件移交等环节。

4.1.2 明确业务目标

高效执法目标、信息共享目标、公众服务目标。

4.2 确定影响颗粒度的关建因素

4.2.1 业务功能复杂性

对于简单功能，可以考虑合并到一个组件中，如将警员信息查询和警员排班信息展示合并为一个“警员管理”组件；对于复杂功能，如案件综合分析，可能需要进一步细分，如将线索整合、数据分析、结果呈现等分别作为小组件。

4.2.2 复用性需求

如身份验证功能，在治安、刑事、交通执法等警务场景均需用到，可作独立组件复用。复用若跨地区，像全国追逃系统的在逃人员信息查询功能，为适配各地接口与规则差异，组件颗粒度应设计得更精细。

4.2.3 数据关联性

紧密关联：在案件侦查过程中，案件线索管理和嫌疑人信息管理功能都与案件数据库紧密相关，这些功能可以考虑放在一个较大颗粒度的“案件侦查数据管理”组件中，以减少数据交互的复杂性。

独立数据：警务装备管理功能主要涉及装备的采购、库存、维护等信息，与案件和人员数据相对独立，可划分成独立的组件，避免与其他组件的数据干扰。

4.2.4 业务变化频率

治安巡逻任务分配规则可能根据季节、治安形势等因素经常变化，对于这种业务变化频繁的功能，应采用较小的颗粒度进行组件化，以便快速修改任务分配算法等内容。如果一个业务变化会导致多个功能的联动修改，如法律条文更新可能影响案件侦查、治安处罚等多个环节的业务规则应用，那么这些相关功能可能需要适当整合为一个组件，方便统一更新。

4.3 建立颗粒度评估指标体系

4.3.1 功能完整性指标

4.3.1.1 业务功能覆盖度

评估低代码组件所涵盖的警务业务功能占全部相关业务功能的比例。对于每个业务功能点，检查组件是否提供了足够详细的操作支持。

4.3.1.2 功能深度

衡量组件对于单一业务功能的处理精细程度。以人员信息查询组件为例，不仅要能查询基本身份信息，如姓名、身份证号等，还应能深入查询关联信息，如犯罪记录、亲属关系、出入境记录等。可以根据能够查询的信息层次数量来评估，层次越多，功能深度越强。

4.3.1.3 功能准确性

通过测试用例来验证组件功能的准确性。准确性可以通过错误率来衡量，即错误操作或错误数据产生的次数占总操作次数的比例。

4.3.2 可维护性指标

4.3.2.1 代码可读性

检查低代码组件代码，看结构是否清晰、变量命名是

Copyright © 2025 by authors and Viser Technology Pte. Ltd.

否合理、注释是否完整。虽其代码量少，但良好可读性利于后续维护与修改，可经代码审查评估，关键变量、函数命名清晰，注释阐释主要功能与逻辑，越符合编写规范，可读性越强。

4.3.2.2 修改便利性

业务需求变化时，考量组件修改难易，以巡逻路线规划组件为例，遇巡逻区域、重点改变，从修改耗时、代码量及技术难度评估能否便捷调整路线参数，若仅微调参数、无需复杂代码重构，修改便利性便高。

4.3.2.3 更新兼容性

考虑组件更新与其他组件、系统兼容性，版本更新（安全补丁或功能增强）时，测试是否影响集成组件（数据共享、报表生成等）正常运行。若组件高度耦合，像报警受理组件与多组件紧密交互且缺清晰接口文档，一旦出故障（如报警信息丢失），定位难度便大增。

4.3.3 组件复用性指标

4.3.3.1 复用频率

记录组件一定时段（月或季）复用次数，复用多即通用性、实用性强，像身份验证组件在接处警、户籍管理常被用，复用频率高。可分等级，月超 100 次为高频，10～99 次中频，不足 10 次低频。

4.3.3.2 复用范围

统计组件在多警务场景复用情况，如数据加密组件，用于内部存贮与外部交互加密等，依复用场景数量、类型打分，越广则分越高；同时考量组件于不同警种（刑侦、治安、交通等）业务复用可能，像地图定位组件助交警定位事故、治安追踪嫌疑人，复用范围广。

4.3.4 性能效率指标

4.3.4.1 响应时间

测量组件执行操作（查询、提交、更新等）的响应时间，警务实时性强业务（如指挥调度位置更新组件）要尽量缩短，用性能测试工具测平均响应时间（以毫秒计），还测高并发场景（大型活动人员信息查询组件），超 5 秒或影响执法。依业务设响应阈值，像紧急报警处理组件须 3 秒内，超时会致业务延迟。

4.3.4.2 资源占用率

评估组件运行时的 CPU 使用率、内存占用等系统资源占用情况，多组件同运行的警务系统里，过高会致卡顿、崩溃，可用监控工具获取数据，如大数据分析组件，运行时 CPU 平均 30%、内存占 2GB，对比系统总量，判断占用是否合理。

4.3.5 开发效率指标

4.3.5.1 组件开发时间

用项目管理工具记录“报警信息处理”组件开发各环节（需求分析、代码编写、测试、文档编制）耗时，对比开发粗颗粒度多功能报警组件与细颗粒度分功能（接收、分类、初步处理）组件的时间差别。

4.3.5.2 组件集成难度

衡量将一个新开发的“移动警务终端数据同步”组件集成到现有警务系统中的难易程度。如果该组件需要对多个现有组件的接口进行适配，并且要进行复杂的数据转换，如将移动终端的数据格式转换为后台系统的格式，那么其集成难度就较高。

4.3.6 组件安全性指标

4.3.6.1 数据安全

评估组件的数据保护能力，涵盖加密、访问控制，像存敏感警务信息（卧底警员信息）的组件，看加密算法强度与访问权限设置合理性，采用如 AES - 256 位加密且严格划分访问权限，数据安全性便高。

4.3.6.2 安全漏洞风险

定期扫描组件安全漏洞，统计数量与严重程度（如 SQL 注入、XSS 等），用专业工具扫描，依出现的中、低风险漏洞数量评估其安全风险。

4.3.7 组件接口标准化指标

4.3.7.1 接口规范性

检查组件接口是否合行业或内部规范（参数命名、数据类型、调用方式等），警务系统集成时，依接口文档对比评估，合规参数占比越高，规范性越强。

4.3.7.2 接口兼容性

验证组件接口与他组件兼容性，多组件集成时这至关重要，如视频监控与分析组件接口对接，要实现数据流畅传输分析。经接口集成测试评估，以测试成功场景数占总测试场景数比例衡量兼容性。

4.4 构建层次分析法（AHP）模型

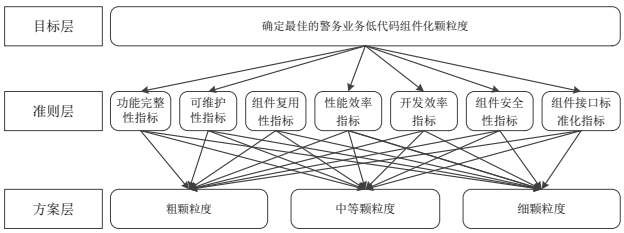

图 2 构建层次分析法（AHP）模型

4.4.1 确定决策目标与层次结构

4.4.1.1 目标层

确定最优的警情处置低代码组件化颗粒度方案，旨在平衡组件通用性、业务契合度与开发成本，提升警情处置效率与系统灵活性。

4.4.1.2 准则层

复用性准则：衡量组件在不同警情类型（刑事、治安、求助等）、辖区场景（城市繁华区、城乡结合部、乡村）及警种（刑警、交警、治安警）应用的重复使用频次与适用性，高频复用利于资源节约与快速搭建。

业务耦合度准则：评估组件与警情处置核心业务逻辑、警务规范流程（接警登记、处警调度、现场处置、办结审

Copyright © 2025 by authors and Viser Technology Pte. Ltd.

核）的紧密关联程度，强耦合需精细适配业务，弱耦合可追求通用灵活。

开发维护成本准则：考量组件初始开发人力（程序员工时、算法专家投入）、技术难度（涉及复杂算法如智能警情分类、地理信息集成）及后续迭代（数据更新、功能优化、故障修复）耗费资源。

4.4.1.3 方案层

针对警情处置关键环节提出多种组件颗粒度方案，如细颗粒度（功能高度原子化、单一功能封装）、中颗粒度（适度聚合关联功能）、粗颗粒度（大模块集成多功能，业务针对性强）。

4.4.2 构造判断矩阵

警务专家、一线警员、低代码开发工程师组成专家组，通过两两比较方式，按 1～9 标度法（1 表示同等重要，9 表示绝对重要）对准则层各因素相对目标层重要性打分，构建判断矩阵 A。专家组认为复用性比业务耦合度稍重要，赋值 3；业务耦合度比开发维护成本略重要，赋值 2，得到：

$$A=\begin{bmatrix}1 & 3 & 2\\ 1/3 & 1 & 1/2\\ 1/2 & 2 & 1\end{bmatrix} \quad (1)$$

针对每个准则层因素，对方案层各颗粒度方案两两比较构造判断矩阵。以复用性准则下为例，专家认为细颗粒度在复用场景更多、适用性强于中颗粒度，赋值 3；中颗粒度比粗颗粒度复用稍好，赋值 2，构建矩阵：

$$B_{\text{复用性}}=\begin{bmatrix}1 & 3 & 5\\ 1/3 & 1 & 2\\ 1/5 & 1/2 & 1\end{bmatrix} \quad (2)$$

$$B_{\text{业务耦合度}}=\begin{bmatrix}1 & 1 & 1/2\\ 1 & 1 & 1/2\\ 2 & 2 & 1\end{bmatrix} \quad (3)$$

$$B_{\text{开发维护成本}}=\begin{bmatrix}1 & 5 & 1/3\\ 1/5 & 1 & 1/3\\ 3 & 3 & 1\end{bmatrix} \quad (4)$$

4.4.3 层次单排序及一致性检验

4.4.3.1 计算权重向量

利用方根法计算判断矩阵最大特征根对应的特征向量，归一化后得各因素相对权重。对矩阵 A，经计算复用性、业务耦合度、开发维护成本相对目标层权重分别为 $W_A=[0.5396146,\ 0.1634241,\ 0.2969613]^T$。

算出各准则下方案层相对权重：

$$W_{B\,\text{复用性}}=[0.648329,\ 0.2296508,\ 0.1220202]^T \quad (5)$$

$$W_{B\,\text{业务耦合度}}=[0.5472164,\ 0.1897093,\ 0.2630742]^T \quad (6)$$

$$W_{B\,\text{开发维护成本}}=[0.5815521,\ 0.1094523,\ 0.3089956]^T \quad (7)$$

4.4.3.2 一致性检验

计算一致性指标$CI=(\lambda_{max}-n)/(n-1)$，（$\lambda_{max}$为最大特征根，n为矩阵阶数），查找对应平均随机一致性指标，RI=0.58，计算一致性比例$CR=CI/RI$，若$CR<0.1$，则判断矩阵满足一致性要求。经计算，上述矩阵均通过一致性检验。

复用性 λ_{max}

$$=\begin{bmatrix}1 & 3 & 2\\ 1/3 & 1 & 1/2\\ 1/2 & 2 & 1\end{bmatrix}[0.648329,\ 0.2296508,\ 0.1220202]^T$$

$$=2.993578629$$

业务耦合度 λ_{max}

$$=\begin{bmatrix}1 & 3 & 2\\ 1/3 & 1 & 1/2\\ 1/2 & 2 & 1\end{bmatrix}[0.5472164,\ 0.1897093,\ 0.2630742]^T$$

$$=3.32066158$$

开发维护成本 λ_{max}

$$=\begin{bmatrix}1 & 3 & 2\\ 1/3 & 1 & 1/2\\ 1/2 & 2 & 1\end{bmatrix}[0.5815521,\ 0.1094523,\ 0.3089956]^T$$

$$=3.704629$$

复用性 $CR=CI/RI=\frac{(\lambda_{max}-n)/(n-1)}{RI}=$

$$\frac{(2.993578629-3)/(3-1)}{0.58}=-0.00321<0.1,$$

业务耦合度 $CR=CI/RI=\frac{(\lambda_{max}-n)/(n-1)}{RI}=$

$$\frac{(3.32066158-3)/(3-1)}{0.58}=0.05366<0.1,$$

开发维护成本 $CR=CI/RI=\frac{(\lambda_{max}-n)/(n-1)}{RI}=$

$$\frac{(3.704629-3)/(3-1)}{0.58}=-0.16864<0.1,$$

经计算，上述矩阵均通过一致性检验。

4.4.4 层次总排序及综合评价

4.4.4.1 计算综合权重

将方案层在各准则下权重向量与准则层相对目标层权重向量加权组合，得方案层各颗粒度方案相对目标层综合权重。

设$W_{综合}=[W_{细},\ W_{中},\ W_{粗}]^T$

$$W_{细}=0.53961460\times0.648329+0.1634241\times0.5472164+0.2969613\times0.5815521=0.611975$$

$$W_{中}=0.53961460\times0.2296508+0.1634241\times0.1897093+0.2969613\times0.1094523=0.187429$$

$$W_{粗}=0.53961460\times0.1220202+0.1634241\times0.2630742+0.2969613\times0.3089956=0.200596$$

4.4.4.2 比较综合权重大小，权重最大者对应颗粒度方案最优

$$W_{综合}=[0.611975,\ 0.187429,\ 0.200596]^T$$

细颗粒度权重最高，表明在当前警情处置业务与专家认知下，细颗粒度组件方案利于低代码平台构建，更契合整体目标。

Copyright © 2025 by authors and Viser Technology Pte. Ltd.

4.5 模型验证与优化

4.5.1 案例验证

选择实际的警务业务场景进行验证，应用评估指标体系对系统进行评估。比较三种方案下系统的可维护性、复用性、性能和开发效率等指标。如在实际运行一段时间后，统计粗颗粒度方案下系统故障定位的平均时间、中等颗粒度方案下组件的复用频率等数据。

4.5.2 反馈与优化

根据案例验证的结果，收集反馈信息。如果发现某个评估指标的权重设置不合理，或者模型没有考虑到某些重要因素，如在验证过程中发现数据安全因素对组件化颗粒度有重要影响，及时调整模型。

4.5.3 持续改进

随着警务业务的发展，如新技术在警务中的应用（如人工智能辅助侦查技术）、新的执法要求的出现，不断更新和完善分析模型。重新审视模型中的关键因素和评估指标，确保模型始终能够准确地指导低代码组件化颗粒度的确定。

5 结论

基于警务业务的低代码组件化颗粒度分析意义重大，应用低代码开发平台与组件化理念，能助警务信息化建设提效。分析时要综合考量性能、可维护、可扩展与开发效率等因素，合理界定组件颗粒度。未来，随信息技术发展，其在警务业务应用会更广，颗粒度分析也将持续完善，助力警务创新发展。

[参考文献]

[1]张宇，王强. 基于组件化的警务系统开发实践与思考[J]. 信息通信技术与政策，2020(8)：66-70.
[2]陈晨，赵亮. 低代码开发平台的组件化设计与实现[J]. 软件导刊，2019，18(7)：23-26.
[3]刘峰，马杰. 警务业务系统中组件复用性的研究与应用[J]. 中国安防，2022(4)：75-78.
[4]王浩，李阳. 基于层次分析法的警务系统组件化评估模型研究[J]. 科技管理研究，2020，40(12)：201-206.

作者简介：张铁龙（1981.11—），男，河北省保定市人，汉族，专科学历，任职于河北雄安新区公安局，从事公安信息化相关工作；姚叙辰（1998.12-），男，河北省保定市人，汉族，本科学历，任职于河北雄安新区公安局，从事公安信息化相关工作；李巧霞（1997.5—），女，内蒙古乌兰察布市人，汉族，本科学历，就职于北京安融畅信息技术有限公司，从事软件开发相关工作；赵全举（1985.11—），男，河北石家庄人，汉族，本科学历，就职于北京安融畅信息技术有限公司，从事软件开发相关工作；杜林佳（1987.7—），男，河北石家庄人，汉族，研究生学历，中级工程师，就职于北京安融畅信息技术有限公司，从事软件开发、警务技术、智能交通相关工作。

Copyright © 2025 by authors and Viser Technology Pte. Ltd.