

# Chaining Geographic Information Services with Kepler-workflow

ZHANG Jianbo<sup>1,2</sup>, LIU Jiping<sup>2</sup>

1. School of Resource and Environmental Science

Wuhan University

Wuhan, China

[finecho@163.com](mailto:finecho@163.com)

2. Research Center of Government GIS

Chinese Academy of Surveying & Mapping

Beijing, China

Wang Bei<sup>3</sup>, GUO Xin<sup>2,4</sup>

3. Institute of Geographic Sciences and Natural Resources Research

CAS

Beijing, China

4. School of Mathematics, Physics and Software Engineering

Lanzhou Jiaotong University

Lanzhou, China

**Abstract**— We study on the patterns of spatial information services chaining, and design an aggregation model based on WPS simple services chaining. By analyzing the main problems between the interface and data control about OWS services used in the workflow, we present the corresponding methods of the interface improvements, and GML-compressed scheduling strategy. Finally, combined with Kepler graphic-workflow, processing flow modeling with OWS services chain and solutions are presented in the paper.

**Keywords**- OWS; combining; services chaining; GML data flow dispatch; graphic-workflow;

## 基于 Kepler 工作流的 OWS 服务链应用

张建博<sup>1,2</sup>, 刘纪平<sup>2</sup>, 王蓓<sup>3</sup>, 郭欣<sup>2,4</sup>

1. 武汉大学资源与环境科学学院, 武汉, 中国, 430079

2. 中国测绘科学研究院, 北京, 中国, 100830

3. 中国科学院地理科学与资源研究所, 北京, 中国, 100101

4. 兰州交通大学数理与软件工程学院, 兰州, 中国, 730070

1. [finecho@163.com](mailto:finecho@163.com), 2. [liujp@casm.com.cn](mailto:liujp@casm.com.cn)

**【摘要】**研究了网络环境下空间信息服务链模式, 给出了改进 WPS 简单服务链的聚合模型。通过分析 OWS 服务链应用于工作流在接口、数据层面的主要问题, 提出了相应的接口改进方法, 以及基于地址引用、压缩 GML 的数据调度策略, 并结合 Kepler 图形工作流, 给出了 OWS 服务链的过程建模。

**【关键词】**OWS; 聚合; 服务链; GML 调度; 图形工作流

## 1 引言

基于工作流技术聚合标准的空间信息服务, 为网络环境下分布式空间信息的共享和互操作提供了有效的解决方案<sup>[1,2]</sup>。然而由于空间信息处理的复杂性、耗时性和异构性等特点, 使得工作技术并不能很好的应用于 GI (geographic information) 领域<sup>[3]</sup>。主要体现在 (1) 接口层面: 面向服务的工作流引擎中执行的服务节点都采用 OASIS<sup>[4]</sup> 标准, 而空间信息服务则采用 OWS (OGC Web Service) 标准, 导致空间信息服务在工作流中不能被聚合 (2) 数据层面: OGC 标准化的 WFS、WPS 服务以 GML 为共享交换格式, 1 密

集的 GML 操作以及长事务处理, 导致工作流执行空间信息服务链效率低。

目前, 对基于工作流的空间信息服务链聚合研究主要集中在两个方面: 从抽象空间信息服务描述中自动派生工作流描述语言 (BPEL), 送入工作流引擎执行<sup>[5-8]</sup>, 以及采用语义、本体的方法构建服务链, 驱动工作流<sup>[9-11]</sup>。这些方法主要是从概念层次上讨论工作流, 并没有针对 GI 领域的空间信息处理服务在工作流中如何建模作深入的探讨。因此, 本文主要研究如何改进 GI 服务接口, 以及服务链执行策略, 使 GI 服务链无缝地应用于图形工作流。

## 2 GI 服务链与工作流

### 2.1 GI 服务选择

基金项目: 国家自然科学基金 (40901195)

OGC 标准化了多种的空间信息服务：地图服务（WMS、WCS）、要素服务（WFS）、处理服务（WPS）<sup>[1, 2]</sup>等。本文以 WFS 要素服务为数据源，WPS 处理服务为节点，构建 GI 服务链。

## 2.2 GI 服务链模式

ISO19119 提出了三种服务链模式：用户自定义服务链模式、 workflow 管理服务链模式以及聚合服务链模式。

用户自定义服务链模式也称“透明”服务链，其客户端应用程序执行用户交互的所有客户端响应编码以及业务逻辑编码，包括服务请求、有效性验证、响应服务返回、数据解析、GML 数据中间存储、异常捕捉。

聚合服务链是一种“不透明”服务链，可以作为单独的服务存在，实现所有 GI 服务的实例化以及聚合过程。用户不了解服务链内部的细节，同时也无法监测单个 GI 服务的执行。

workflow 管理服务链模式是一种“半透明”服务链，用户不再实现具体服务的执行过程，仅仅将关注的重点集中在业务逻辑上，因此实现了业务逻辑定义和执行过程的分离。

“透明服务链”是一种硬编码的方式，代码重用性低。“不透明服务链”屏蔽所有服务细节，用户无法定义业务逻辑，粗粒度的服务聚合缺乏灵活性。因此，只有“半透明服务链”能够按照用户的需求动态构建 GI 领域问题模型，是一种相对灵活的服务链聚合方式。

## 2.3 工作流选择

“透明服务链”的模式，即 workflow 驱动服务链模式。本文认为核心的问题在于①所选 workflow 接口如何支持 GI 服务链执行②所选 workflow 如何调度 GML 数据，改善 GI 服务链的执行效率。

workflow 引擎按照过程建模的不同可分为两种：BPEL 过程建模 workflow 和图形建模 workflow。基于 BPEL 过程建模的 workflow，采用 BPEL 语言谱写 workflow，然后交给 workflow 引擎执行，但是 BPEL 语言的谱写缺乏可视化工具，谱写困难。

图形建模 workflow，就是用户通过拖拽图形代理的方式定制服务链。相比 BPEL 过程建模 workflow，具有以下优点：①提供了过程建模的可视化工具，用户不再通过手工谱写过程建模语言。②客户端应用程序只需要实现 WMS、WFS 要素服务的可视化，执行过程交由图形 workflow 执行，实现应用程序客户端、图形 workflow 客户端以及 workflow 引擎最大程度上的松耦合架

构。③经过接口改造的图元代理，能够动态发现、聚合 GI 服务，易于实现 GML 数据调度策略（3.1 阐述）。

因此，本文选择图形 workflow 聚合 GI 服务链。

## 3 图形 workflow 驱动 GI 服务链模型

### 3.1 问题域与概念模型

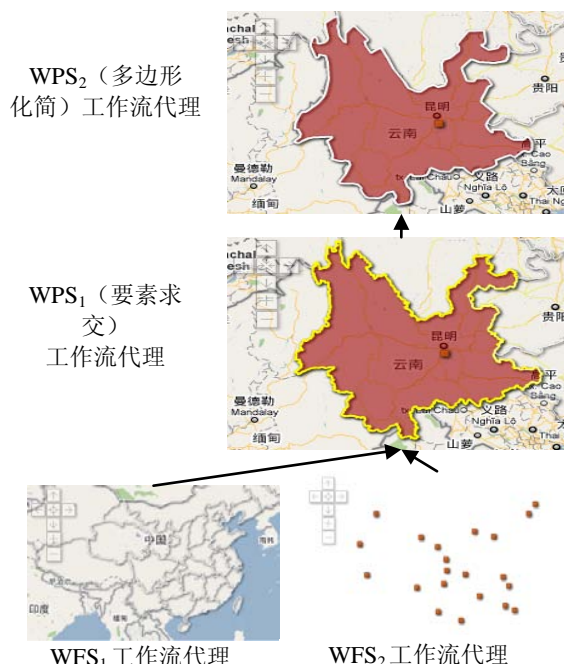


Fig.1 conceptual workflow model

图 1 工作流概念模型

拟解决以下问题域：展示全国某机构办事处所在省份的多边形要素。由于瘦客户端加载过多的多边形要素，其显示速度极慢，需要对空间要素进行多边形化简。涉及的 GI 服务均为 OGC 标准的空间信息服务具体如下：

- 100W 中国省级行政区划要素服务（WFS1）
- 某机构办事处的点状要素服务（WFS2）
- 要素求交服务（WPS1）
- 要素化简服务（WPS2）

图 1 说明了图形 workflow 聚合上述标准 GI 服务步骤。在图形 workflow 中，workflow 代理节点首先获取 WFS1、WFS2 矢量要素，通过 WPS1 处理服务得到要素相交的多边形要素，由 WPS2 对多边形化简，满足瘦客户端的可视化要求。后续文章以此概念模型阐述接口、数据调度以及图形 workflow 的过程建模。

### 3.2 改进 WPS 服务链的模型描述

WPS 标准定义了简单的服务链。要求服务链中参与聚合的 WPS 服务实例以 KVP 编码的方式提交

GET 请求。可以描述为:

GetURL=req(req1(req2(reqn(…)))) (1)

其中 req 为当前 WPS 服务请求, req1 为参与服务链的 OWS 服务请求, 包括 WFS 的 GetFeature 以及其它 WPS 的 Execute。从 (1) 中可看出, WPS 标准定义的服务链以嵌套 URL 的方式聚合服务请求, 由于 URL 编码的字符限制, 这种服务链仅仅适合少数 OWS 服务的聚合; 并且嵌套的服务聚合给服务监控以及异常捕捉带来很大的困难。

图形 workflow 引擎聚合 OWS 服务, 不再采用 KVP 编码接口 (3.1 详细描述) 如 (2) 描述:

V=Workflow(OWS1, OWS2, ..., OWSn) (2)

不再以 WPS 为单独的服务链入口, 而是将嵌套的服务拆分为多个 WFS、WPS 服务, 以类似流水线式的方式通过图元代理聚合服务, 如图 1。工作流中的 OWS 服务通过多点服务增值的方式聚合, 而不是 (1) 描述单点 WPS 服务聚合; 语义层面的 URL 聚合变为 GML 格式的数据体聚合。

以 WFS、WPS 标准为基础, 本文采用图形 workflow 聚合 GI 服务, 遵照概念 workflow 模型, 主要 workflow 为: ①首先对标准空间信息服务 WFS、WPS 添加 WSDL 描述, 对消息响应采用 SOAP 绑定; ②改进 WFS 接口, 使之能够输出 GML 地址引用; 为 WPS 接口添加接口解析器, 便于 WPS 非 GML 数据类型的参数传递; ③客户端构建 GI 服务链流程, 采用“流式”方法调度数据, 并由图形 workflow 引擎完成执行、监测、输出工作。

#### 4 图形 workflow 驱动 GI 服务链实现

本文使用 Kepler<sup>[14]</sup>作为图形 workflow 引擎, 根据图形 workflow 以及标准空间信息服务的特点, 通过分析 GI 服务链面向图形 workflow 的接口以及数据调度问题, 提出图形 workflow 引擎聚合 GI 服务链的执行策略。

##### 4.1 WFS、WPS 接口封装

GI 服务与图形 workflow 在内部接口和外部接口层面都存在较大的差异:

1) 外部接口: ①Kepler 工作流面向标准 Web 服务。节点映射的服务以标准的 WSDL 描述, 节点之间以 SOAP 为通信协议进行互操作; 但是 OGC 并没有给出相关规范; ②虽然 OGC 在 OWS-5 中, 提出了 WPS 要支持 WSDL、SOAP 协议的需求, 但是没有明确提出支持 WFS 的 WSDL 封装, 因此图形 workflow 不能协调 WFS、WPS 的服务聚合。

针对这种外部接口层面的不一致问题, 需要针对

WFS、WPS 进行接口封装。以 WPS 为例: 包含三种接口 GetCapabilities、Execute、DescribeProcess。仅需要将 GetCapabilities 请求通过 WSDL 封装, 同时改造 Execute 请求, 使之支持 SOAP 协议, 如图 2 所示。那么 kepler 引擎驱动 WPS 服务能够从封装后的 GetCapabilities 中获取 Execute 的 SOAP 请求。

```
<wsdl:definitions name="SpatialAnalysis" >
<wsdl:types>
<xsd:schema elementFormDefault="qualified" >
<xsd:element name="ExecuteProcess_GMLBufferResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="ExecuteProcess_GMLBufferResult" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="ExecuteProcess_GMLBufferSoapRequest">
<wsdl:part name="parameters" element="tns:ExecuteProcess_GMLBuffer" />
</wsdl:message>
```

Fig.2 WPS with WSDL

图 2 WPS WSDL 描述

2) 内部接口: ①虽然为 WPS 添加了 WSDL 描述, 但是 WPS 的输入参数类型并不在 GetCapabilities 接口中描述, 而是在 DescribeProcess(WFS 为 describeFeatureType)接口中定义, kepler 工作流不能够从 GetCapabilities 接口中提取 SOAP 请求所必备的参数类型; ②OGC 标准没有针对 WPS 的输入输出参数做强制规范, 可以是单独参数也可以是复杂类型 complexType 的多参数输入输出, WPS 入口无法动态识别复杂类型参数。

例如图 1 描述的 GI 服务, WPS1 的输出接口 Response1 可能为 GML 实体数据类型, 也可能是包含在 GML 中的 URL 地址引用类型。那么, 在 Kepler 驱动的这个 GI 服务链中, WPS2 输入接口无法确认是 GML 实体数据, 还是 URL 地址引用, 因为两者的参数类型都是 AnyURL 这一弱类型语言。这一问题的实质是图形 Kepler 工作流无法自动解析非实体化 GML 数据。



Fig.3 WPS output interface define

图 3 WPS 输出接口解析器

解决这个问题需要在 WFS、WPS 服务的输出端, 添加 GML 解析器, 如上图所示: 假设 WPS1 输出 GML, 非 LineString 类型, 而是包含遥感图片的 URL 地址, 那么 GML 解析器就为下一节点 WPS2 解

析出正确的 URL 地址及其类型，而不是 GML 数据体。需要指出的是，GI 服务链构建者不需要另行编码设计，Kepler 提供了 XML 解析组件，用户需要利用这种组件提取所需的参数以及类型。

## 4.2 基于地址引用、压缩 GML 的数据调度策略

标准 GI 服务链，以 GML 为共享数据交换格式，密集的 GML 输入输出给 workflow 驱动 GI 服务链带来一定的问题：①用户请求 WFS 服务，并将其下载到客户端，需要额外的网络传输时间开销；②网络传输大数据量的 GML 速度慢；

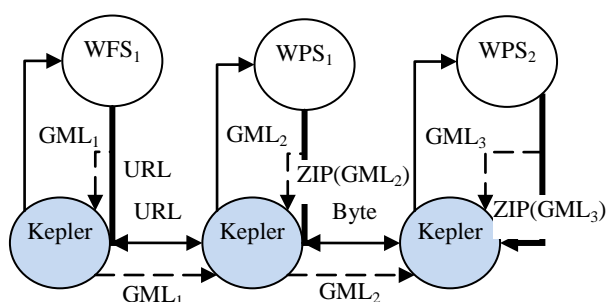


Fig.4 data flow of GI services chaining  
图 4 图形工作流模式下服务链数据流图

图 4 表达了数据调度的两种策略。细实线为 kepler 代理的服务请求，虚线和粗实线表示数据传输的两种方式，数据请求、传输由各个图元节点控制。

传统的数据调度方法如虚线所示，所有的 OWS 服务都返回 GML 数据，是一种高密度 GML 调度模式。模拟数据传输过程可用以下方法：

$$GML1 = \text{KeplerProxy}(WFS1:\text{GetFeature}) \quad (3)$$

$$GML2 = \text{KeplerProxy}(WPS1:\text{Execute}(GML1)) \quad (4)$$

$$GML3 = \text{KeplerProxy}(WPS2:\text{Execute}(GML2)) \quad (5)$$

由于从一个 OWS 服务到下一 OWS 服务包含服务响应和数据传输两个环节，其总体时间消耗为：

$$T = 2O(N1) + 2O(N2) + O(N3) \quad (6)$$

其中，N1、N2、N3 分别为 GML1、GML2、GML3 的数据量。

为了减少 GML 在网络的传输次数，本文改进了 GI 服务链的调度策略，如上图粗实线所示：①将 WFS1 的 GML 实体输出类型改造为包含 URL 的 complexType 类型；②将 WPS 的 GML 实体输出类系改为压缩包的参数类型，并通过二进制流的方式在网络上传输；③为 WFS、WPS 添加接口解析器（3.1），保证改进后的参数类型能够被图形工作流正确解析。其过程可用下面公式表达。

$$URL = \text{KeplerProxy}(WFS1:\text{GetFeature}) \quad (7)$$

$$ZIP(GML2) = \text{KeplerProxy}(WPS1:\text{Execute}(URL)) \quad (8)$$

$$ZIP(GML3) = \text{KeplerProxy}(WPS2:\text{Execute}(ZIP(GML2))) \quad (9)$$

在数据流调度中，由于为 Kepler 添加接口解析器（3.1），WPS1 可以接受 WFS1 传递来的 URL。那么 workflow 引擎就不必先将 GML 下载到本地，然后传输给 WPS1，而是将响应结果作为 URL 直接交给 WPS1 处理。本质上是避免了 GML 到代理客户端的响应传输（URL 的传输代价忽略不计）；同时，WPS1 可以以 ZIP(GML) 的格式输出 GML，最大限度的压缩 GML。整个服务链的传输时间消耗为：

$$T = O(N1) + 2O(K*N2) + O(K*N3) \quad (10)$$

其中 K 为 GML 的压缩率。由于  $K*N < N (0 < K < 1)$ ，相比（6）其时间消耗明显减少。

## 4.3 Kepler 过程建模

在提出图形工作流中 GI 服务链的接口改进方法以及调度策略的前提下，本文提出面向 kepler 工作流的 GI 服务链过程建模。

图 5 描述了采用 kepler 引擎聚合 GI 服务链的控制流模式图。在 kepler 中，面向 Web 服务的图元节点称为 Actors，实质上是服务代理客户端，通过简单的插件机制执行任何 WSDL 定义的 Web 服务，并通过其端口链接到其他 Actors。利用 kepler 的图形界面定制工作流，我们首先初始化所需要的服务代理 Actors，按照 WFS、WPS 服务的参数要求分别配置代理端口，然后按照 GI 服务链的业务执行逻辑连接 Actors，组成工作流。其数据流向按照图 4 中的描述进行映射。

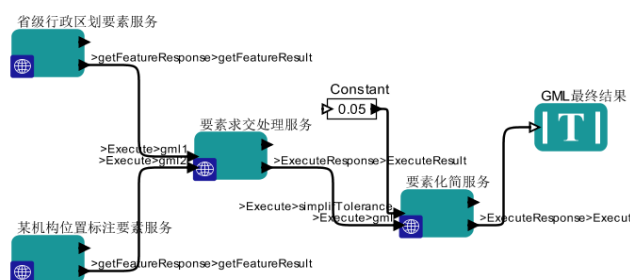


Fig.5 Kepler workflow model  
图 5 kepler 过程建模

接口改造①首先在服务端按照 3.1 方法对上述 OGC 服务进行接口改造，使之满足图形工作流对 GI 服务链采用 WSDL 描述的要求。此时的 WFS 以及 WPS 的 GetCapabilities 接口作为 kepler 节点映射服务的 URL 地址，如：<http://localhost:35830/wps%E8%A>



F%B7%E6%B1%82%E4%B8%8E%E7%9B%B8%E5%BA%94%EF%BC%88sharp%EF%BC%89/WSDLServices/WebService.asmx?wsdl。②改造 WFS 的输出接口，输出的 GML 文件不再表达矢量对象，而是包含 AnyURL 类型的 URL 地址引用。实质上是将 WFS 逻辑对象转换成地址引用，减少 GML 操作。③为 WPS 接口添加接口解析器，解析 WFS 的输出文件中包含的 GetFeature 请求返回的 URL 引用地址。然后交给下一节点的 WPS 执行。

服务发现：kepler 引擎提供 WEB 服务搜索工具-图元代理 Actor，可通过拖拽的方式将参与 GI 服务的 Actor 代理工具置于 workflow，actor 代理按照 WSDL 描述自动发送 SOAP 协议，并捕获输出以及异常。

服务聚合：①配置图元代理 Actor1、Actor2，以发现 WFS1、WFS2 要素服务，发送 GetFeature 请求（SOAP 协议），返回 GML1、GML2，但包含的是中国省级行政区划要素、某机构位置要素的 URL 引用；②将 GML 通过解析器解析出要素地址引用；③图元代理 actor3 接收 URL 引用，并发送 Execute 请求，内部调用 WFS 的 GetFeature 请求，获取参与求交运算的 GML 数据；然后通过矢量求交运算处理，获取各个某机构位置、所在的省的多边形混合数据 GML3；④最后交由面状要素化简处理服务化简数据，得到 GML4，交由客户端验证显示。

可见，图形 workflow 引擎，能够为各种标准 OGC 服务提供请求、执行以及聚合代理，并且按照上述接口以及数据调度方法执行 GI 服务链，业务逻辑性清晰、执行效率高。

## 5 结论

上述文章利用图形 workflow 为聚合标准 OWS 服务提供了新的解决方案：①突破了 GI 服务应用于 workflow 的接口瓶颈②改进的数据流调度策略提高 GI 服务

链执行效率③为聚合 OWS 服务提供应用示范。本文没有针对大数据量的 GI 服务聚合展开讨论，将是下一步研究的重点。

## References (参考文献)

- [1] Alameh N, Chaining Geographic Information Web Services[J], IEEE Internet Computing, 2003,7(5),P22-29
- [2] Alameh N, Service Chaining of Interoperable Geographic Information Web Services[J], IEEE Internet Computing, 2003, 7(5):22-29.
- [3] Friis-Christensen A, Lucchi R, Lutz M, et al, Service chaining architectures for applications implementing distributed geographic information processing[J], International Journal of Geographical Information Science, 2009, 23(5):561-580
- [4] OASIS, 2006, Reference model for service oriented architecture 1.0. Technical report, Organization for the Advancement of Structured Information Standards.
- [5] Jia Wenyu, Li Bin, Gong Jianya, Research on Dynamic GIS Chain Based on Workflow Technology[J], Geomatics and Information Science of Wuhan University,2005,30(011):982-985.  
贾文珏, 李斌, 龚健雅. 基于 workflow 技术的动态 GIS 服务链研究[J]. 武汉大学学报: 信息科学版, 2005, 30(011):982-985.
- [6] Granell C, Gould M, Ramos F. Service Composition for SDIs: integrated components creation[J]. 2005.
- [7] Kiehle C, Greve K, Heier C. Standardized geoprocessing-taking spatial data infrastructures one step further[A]. CiteSeer.2006.
- [8] Kiehle C, Greve K, Heier C. Requirements for next generation spatial data infrastructures-standardized web based geoprocessing and web service orchestration[J]. Transactions in GIS, 2007, 11(6):819-834.
- [9] Lemmens R, de By R, Gould M, et al. Enhancing geo-service chaining through deep service descriptions[J]. Transactions in GIS, 2007, 11(6):849-871.
- [10] Lemmens R, Wytzisk A, de By R, et al. Integrating semantic and syntactic descriptions to chain geographic services[J]. IEEE Internet Computing, 2006, 42-52.
- [11] Friis-Christensen A, Ostl nder N, Lutz M, et al. Designing service architectures for distributed geoprocessing: Challenges and future directions[J]. Transactions in GIS, 2007, 11(6):799-818.
- [12] OGC, 2007b, Web processing service v1.0.0. Implementation specification OGC 05-007r7, Open Geospatial Consortium Inc.
- [13] Yang C, Wong D, Yang R, et al. Performance-improving techniques in web-based GIS[J]. International Journal of Geographical Information Science, 2005, 19(3):319-342.
- [14] Kepler: An Extensible System for Scientific Workflows, <http://kepler.ecoinformatics.org>.