

面向主题的演化业务信息系统

Stephan Schiffner, Thomas Rothschädl, and Nils Meyer

Metasonic AG, Pfaffenhofen, Germany

摘要：现今快速变化的业务环境使得公司的业务和相关信息系统处于不断的变化当中。业务信息系统的演化是处理利益相关者提出与目标的内容和行为相关的部分改变的应用，以对齐新的业务需求。一种处理应用行为变化的方法就是通过改变潜在的业务流程。**S-BPM** 实现了一种方法，其中流程模型可被合适的工作流引擎解释且使利益相关者可以直接执行其上的一般应用。这样，这个一般的应用可视为主系统，在其上可以通过编辑流程模型来进行再设计。本文将演化的业务信息系统与 **S-BPM** 进行比较以提出实现基于 **S-BPM** 的演化业务信息系统的软件需求，其中系统的行为就是潜在业务流程持续演化的结果。

1 引言

现今的企业处于动态的世界当中，常面临新的风险和机遇。只有快速主动响应变化环境，组织才能有强大的竞争优势。据 2012 年 IBM 全球 CEO 面谈调查，参与调查的 CEO 有 1700 多个。调查显示公司必须改进对变化市场需求的响应时间。另外，2009 年经济学人上的一则报道显示世界上接受采访的 CEO 中，90% 的人认为在现今快速变化的业务环境中组织的敏捷是核心的砝码。

在企业中，业务流程驱动着企业的运作，因此流程敏捷是组织敏捷的基础。在 Gartner 报告中，业务分析师将流程敏捷定义为：组织感知需要变化和及时响应变化做出改变运作的的能力。运作的改变将设计工作人员行为的改变和应用程序的改变。

典型地，业务分析师尝试建模业务流程来探索问题，该流程包含所有可能的 if-then-else 工作流路径来确定公司员工要执行的工作顺序。这种方法的副作用就是：尽管可以预定义业务流程的所有复杂情况，但同时业务可能已经演化了然后迫使分析师需要从头开始。一个不同的方法是在短时间内捕获业务流程中最明显的部分，然后将识别并定义不可预测的路径作为业务的一部分。

S-BPM (Subject-oriented Business Process Management) 是一种建模业务流程的方法。**S-BPM** 模型可直接被合适的工作流引擎解释，建模语言（由 5 个符号组成）的简单语义满足利益相关者执行建模任务的最小化需求。

处理变化的业务需求的另一个方法是由 Neumann 等人提出的演化业务信息系统 **EBIS** (Evolutionary Business Information System)，它应该提供能够满足增量变化的业务需求的专门信息系统基础设施，这样（未预期）变化可以直接被利益相关者增量地包含进来。在这样一个系统中，终端用户可以在不中断任何服务的情况下改变应用的内容和行为。

将已有的 **S-BPM** 特征与 **EBIS** 属性相比，两者有很大的交集，这使得我们可以假设：**S-BPM** 是一个好的基础来建立演化业务信息系统。因此，在本文中我们连接这两个概念，目标是提出基于 **S-BPM** 实现流程驱动的 **EBIS** 软件需求。

本文介绍了我们进行中的初步研究结果，有助于引导 **S-BPM** 驱动的演化业务信息系统概念设计和开发前的原型设计。

本文组织结构如下：第 2 节深入描述 **EBIS**、**S-BPM** 和终端用户参与之间的连接，展示了这些话题的合并来产生流程驱动的演化业务信息系统；第 3 节导出实现这样一个系统的软件需求；第 4 节展示了我们正在实现的两个原型；第 5 节用未来研究的开放问题概述来进行总结。

2 EBIS 与 S-BPM 的结合

A. 业务流程建模中终端用户的参与

BPM 项目通常由管理人员发起，而建模任务则由建模专家（比如咨询人员或公司专门训练过的员工）来完成。相关的业务部门人员和创建解决方案的终端用户通常只在需求分析时参与进来，比如在研讨会时。该方法导致的一个事实就是：终端用户察觉到流程模型与现实工作实践的不同。而且终端用户的参与需要专家的指导，这就是被称为所谓的引导者瓶颈[1]，从而降低了组织的敏捷度。

这些问题的一个解决方案就是终端用户在建模流程中的参与。

Michael Prilla 和 Alexander Nolte 在这个话题上做了一些研究。在文献[2]中，他们集中解决终端用户如何集成到模型生命周期中和在哪个阶段被已有的方法技术可以很好的支持这个问题。他们显示当前建模任务和流程模型的使用仍然是主要的一个专家任务，而且支持这些议题的方法还有待开发。这些启发对应了文献[3]中给出的结果，他们做了两个实证研究来分析 BPM 如何从 Bottom-up 建模方法中获益的。这项工作的结果是：他们给出了 5 个建议如何在建模活动中集成终端用户，包括使模型对终端用户容易获得和将他们集成到日常的工作实践中。他们也建议提供新颖的工具来简化终端用户在建模流程中的参与，重新定义建模专家的角色到一个更加支持方向，和在 BPM 中开发一个结合 top-down 和 bottom-up 两种策略的方法。在文献[1]中，他们显示：甚至没有被训练的人可以使用模型有合作地协商流程内容，并且为他们模型交互定义最小的需求，特别通过减少使用的元素来特别简化建模语言语义。

简化终端用户建模的工具开发也是惠灵顿维多利亚大学 Pedro Antunes 提出的人道主义业务流程研究议程的一部分。其中的目标之一就是许可终端用户来建模他们自己的流程。当前的做法是：一群人在一个应用上工作，这使用户以一种非正式的方式来建模流程。

Peter Rittgen 特别从团队的角度来研究终端用户建模的影响。他发现没有被训练的终端用户一般有能力来执行建模任务，越高的工具支持，模型的质量就越高。另外，补充的团队相比配套的团队能达到更好的一致。

B. S-BPM

S-BPM 集中关注在业务流程的执行元素。这些执行元素被称为主题，它们将自己的行为（活动）以局部流程模型的形式封装起来。图 1 显示了一个包含两个主题和它们之间消息交换的通信图的例子。

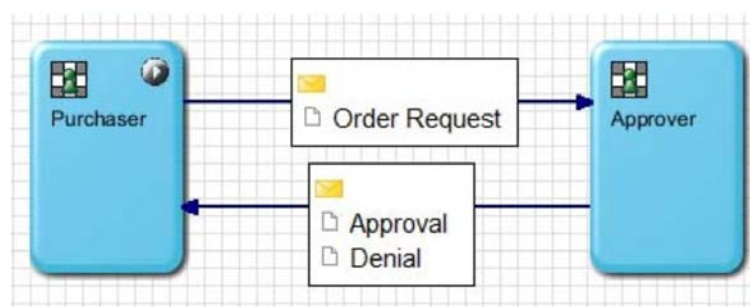


Figure 1. S-BPM 通信图

在 S-BPM 中有 3 种类型的状态。**功能状态**表示一个具体的动作，比如填写主题的一个申请。**发送状态**和**接收状态**表示通过消息交换与其它主题的交互。消息是必要的，用于同步不同主题的行为。执行路径通过模型状态之间的变迁定义在主题行为中。

图 2 显示了主题 Approver 内部行为图的一个摘要，其中 Receive Order 是一个接收状态，Check Order Request 是一个功能状态，Send Approval 和 Send Denial 是发送状态。

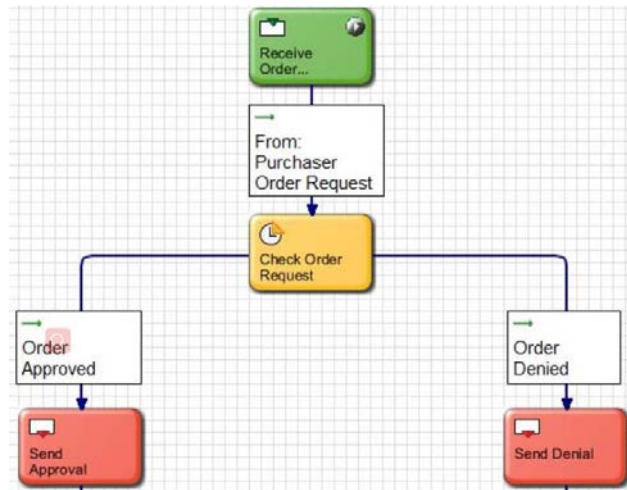


Figure 2. 主题 Approver 的 S-BPM 内部行为

S-BPM 模型使用 5 个符号来进行图形化表示，即 3 个状态类型、消息和主题。此外，文献[4]证明了每个语法正确的 S-BPM 模型均可被合适的工作流引擎所解释。更新的集中在 ad-hoc 改变的 S-BPM 研究工作，也称为动态 S-BPM，可见文献[5]。

C. 演化业务信息系统的属性

Neumann 等人将 EBIS 定义为一个专门的信息系统基础设施，它可以增量地满足变化的业务需求，（未预期）变化可以直接被利益相关者增量地包含进来。他们提出的方法是基于再设计的概念[6]。

Van Aken[7]提出当处理组织开发时应该应用设计科学的概念。再设计的方法是更进一步。这就质疑了“当使用开始设计就结束”这样一种说法，而且认为人与他们的社会和物理环境是不可分离的。因为基本的设计是满足预期的需求，而再设计是针对变化需求的改编[8]。已有的研究表明再设计能够通过再设计阶段植入系统的原则来支持[9]。因此，支持再设计方法的信息系统应该根据系统的内容和行为能够被用户所适应。用户应该成为信息系统再设计的主要执行者。

文献[6]列出了这样一个系统应该有的特性：

- ✧ 与增量变化的无缝集成；
- ✧ 通过多个贡献者的内容和行为再设计；
- ✧ 系统检查以提供关于当前状态、行为和之前动作的反馈；
- ✧ 管理实例和模式的多个版本和变体；
- ✧ 系统/组织需求（治理、稳定性）与单独需求（灵活性、可裁剪性）的平衡。

Neumann 等人分析了已有的系统原型以哪种方式来满足这些需求。文献[6]显示的结果如下表所示。每个属性的细节讨论将在接下来的章节中给出，同时将这些需求与 S-BPM 概念进行比较。

Table 1. 系统原型和它们的演化属性

	Wiki	EM-Systems	Ad-Hoc WF System	Evolutionary Business IS
Ad-Hoc Changes	+	+	+	+
Secondary Design	+	+	-	+
Content Development	+	+	-	+
Instance Development	+	+	+	+
Schema Development	-	-	-	+
Variability Management	+	-	+	+
Control Flow Adaption	-	-	+	+
Feedback Channel	+	-	-	+
Governance Support	-	-	-	+

D. 连接 S-BPM 与 EBIS 的属性

基于 S-BPM 的方法，可实现一个信息系统，其中可以建模面向主题的业务流程并且可以立即被合适的工作流引擎解释。通过提供一般的用户接口应用，利益相关者可以执行他们的工作任务，应用行为定义在当前已解释的流程模型中。我们假设这样一个系统能够形成一个演化业务信息系统的基础。

接下来，我们连接 S-BPM 和上节列出的 EBIS 属性，讨论它们是否被 S-BPM 的概念所支持。

再设计与内容开发

关联基于 S-BPM 的信息系统和演化的业务信息系统，我们认为一般的用户接口应用和工作流引擎可视为主要系统，在其之上通过修改潜在的流程模型执行再设计以达到应用的行为改变。同样，可以改变应用的内容，因它是在 S-BPM 流程模型中用不同的视图来建模业务对象，从这个流程模型中产生对应的输入表单。在基于 S-BPM 流程执行引擎下的运行环境中，关于流程执行时的行为改编和甚至是完全新的流程定义的描述可参见文献[5]。

Ad-Hoc 变化和实例开发

在一个流程实例执行的时候，一些不可预见的事件可能发生，这是在潜在的流程模型中所没有覆盖的。因此，可以让演化业务信息系统的用户立即执行 ad-hoc 变化。实例级别的 ad-hoc 偏离这一话题是 S-BPM 领域中几项研究的焦点，最早的原型可参见文献[5]。

模式开发

在一个演化的业务信息系统中，变化不仅在实例级别，同时也允许在模式级别。这意味着领域专家能够基于已有的模型修改工作流或创建新的工作流。鉴于这一需求，系统需要实现对非建模专家也有能力来持续优化潜在的业务流程。S-BPM 的方法已经可以满足为外行用户参与的最小需求，正如文献[1]中 Nolte 和 Prilla 所描述的，比如：简化建模语言的语义。尽管在已有的 S-BPM 实现当中，一个流程的不同版本可以同时工作，但还不存在可以不中断服务的情况下允许执行这些变化的系统。建模任务必须在单独的环境中执行，并且新的或改编的流程描述必须上传到工作流引擎。而且，也不可能自动化实例迁移。根据 ad-hoc 实例变化，一个增量模式开发的 S-BPM 方法在文献中已有描述（德文）。

可变性管理

如果终端用户自己能够执行实例和模式改编，那么大量的变种可能产生，部分模型也许是可重用的或组合形成新的流程[6]。从利益相关者的角度看，S-BPM 的概念来分割流程到主题产生一个清楚的关注点分离，并且可以基于传入和传出的消息定义结构良好的通信接口，这将促进在新模型中重用的可能性。

控制流改编

由于 S-BPM 是一种创建业务流程的方法，控制流的改编是一个关键特征。既然控制流对主题而言总是本地定义的内部行为，变化的产生不会影响到其他的利益相关者（只要这些变化不需要引起消息交换的改编）。在演化业务信息系统的上下文中，我们认为这是一个优势，既然控制流变化的产生大部分局限在个性化的利益相关者视图中，因此在工作流中对其它利益相关者带来无意的影响的风险将会减轻。

反馈渠道

根据文献[6]，系统的改编必须是可追踪的，而且利益相关者必须获得关于未来预期变化所带来的影响信息。在 S-BPM 中，可追踪性可通过记录变迁和流程实例执行的消息来达成。一旦 ad-hoc 变化产生，执行系统改编的用户必须激活这些功能。变化和激活必须被追踪和分析。达成这一目的的一个方法在已有文献中有讨论（德文）。

治理支持

在演化业务信息系统中，系统改编的可能性应该被限制在用户的责任域中。不同主题相

互通信的概念促进了基于责任的关注点分离来支持这一需求。已有文献中（德文）描述了一个关于如何使终端用户在运行时刻的改编以一种细粒度的方式被约束的方法。

E. 小结

从 D 节的对比总结中可以看出，演化业务信息系统特性能够被使用 S-BPM 的概念所覆盖。对于一些特性已经存在了一些解决方案，对其它的也已经有了原型或概念。

典型的工作流话题，比如控制流改编、实例开发或 ad-hoc 变化均可被很好的支持，并且为实现流程驱动的应用平台提供了基础，允许终端用户的内容开发和再设计。后者被很好的支持，因为：首先，S-BPM 建模语言容易被利益相关者使用；第二，根据相关参与者的责任，主题的概念自动地产生流程模型的分割。

当考虑模型管理和反馈机制相关的任务时，一些开放性的问题被提了出来。尽管 S-BPM 能够支持像治理或可变性概念这样的话题，但是当前很少有对应这些概念的研究。而且，不中断服务的模式开发问题和反馈与预期变化的影响仿真问题仍然还有待回答。

表 2 显示了动态 S-BPM 与文献[6]中提到的原型的关联。在接下来的章节中，我们将深入探索一些关于基于 S-BPM 实现演化业务信息系统的实际问题，并提出相应的软件需求。

Table 2. 与 S-BPM 相关的 EBIS 属性

	Wiki	Ad-Hoc WF System	Dynamic S-BPM	Evolutionary Business IS
Ad-Hoc Changes	+	+	+	+
Secondary Design	+	-	+	+
Content Development	+	-	+	+
Instance Development	+	+	+	+
Schema Development	-	-	0	+
Variability Management	+	+	0	+
Control Flow Adaption	-	+	+	+
Feedback Channel	+	-	0	+
Governance Support	-	-	0	+

3 面向主题的演化业务信息系统的软件需求

A. 需求

授权利益相关者修改信息系统是演化业务信息系统的一个关键需求，在我们的上下文中这意味着改编 S-BPM 流程模型的能力。接下来，我们认为这个需求是后续软件需求的主要驱动之一。在本文的剩余部分，我们使用术语“流程模型”来简化“面向主题的流程模型”。

遵循 S-BPM 的方法，一个流程模型由两个抽象层组成。首先，在通信视图中，映射了哪些参与者涉入到流程中以及如何通过发送和接受消息彼此交互。第二，在所谓的主题内部行为中定义了单个流程参与者的工作流。

授权终端用户执行模型改编导致一个问题：用户应该允许改变流程模型的哪些部分。在演化业务信息系统中，用户应该只允许执行在他的责任域中的改变[6]。既然主题封装了依赖角色的任务，那么流程分解到主题便支持这一需求。除此之外，主题间的通信必须相互协调。尽管终端用户应该有关于对他们工作和对其他参与者（主题）的直接接口有好的了解，我们认为参与主题的选择和他们之间的交互需要更广的流程概览，还有一致性的需求和有策略的公司目标。而且，通信模型的改编可能对流程性能带来不可预见的影响。因此，提出**需求 1**：在面向主题的演化业务信息系统中，必须有正确的机制从两方面来限制建模。第一，负责一个主题的终端用户（主题的所有者）必须仅允许改变他们主题的内部行为；第二，通信模型的改变必须仅限于流程的所有者。

流程通讯模型的改变可能导致被影响主题的内部行为的改变，即引入新的消息。优化主题的内部行为之后，必须验证消息的交换仍然是一致的，意味着所有预期的消息都是主题发送或接收的。为此，提出**需求 2**：在面向主题的演化业务信息系统中，必须实现自动化的验证机制，验证主题行为或通信模型改编后在消息交换上模型的一致性。

一般地,对流程中一个具体主题的角色可能不只有一个具体的用户。一个典型的场景是:部门的多个员工工作于不同的流程实例中,但他们都有同样的角色。如果一个用户修改了流程模型,这将影响到所有其他的用户,既然新的流程实例已经产生在新的模型版本中。如果缺乏通信,可能带来三个潜在的问题:第一,主题的不同用户对于模型的改变不能达成一致;第二,可能带来模型粒度上的问题,意味着初学者相比专家需要更详细流程模型;第三,如果其他用户没有被通知到关于流程的改变,这可能会对已经在运行的流程实例带来负面的影响。因此,我们提出**需求 3**:在面向主题的演化业务信息系统中,必须有组件支持流程模型的协同验证以达到关于对改变的一致同意。另外,我们增加**需求 4**:在面向主题的演化业务信息系统中,应该可以对模型进行订阅,以收到关于改变的自动通知。

流程模型的偏离,要么是暂时性的,仅针对一个具体的实例,要么是永久性的影响模式,针对未来所有的实例。

如果是暂时性的改变,随着时间的过去,对于单个流程可能累积大量版本,这些改变可能有交集,也就是说员工在模型的某个点执行了相似或相等的偏离。这一方面效率可能是低下的,既然多个用户都执行了相同的任务;另一方面这样的交集可能意味着原始流程模型的不足。因此,我们提出**需求 5**:在面向主题的演化业务信息系统中,终端用户应该可以容易地获取其他用户做的模型改编,也能够临时地(ad-hoc)作为构建模块重用它们。

针对永久性的模型改编,必须处理并发带来的冲突,意味着不同的用户并发地工作于同一个模型。在 S-BPM 模型中,改变要么在主题级别要么在通信级别,两者均可能带来重复或冲突的改变。特别地,如果流程跨度多个部门又拥有许多用户,管理变化就是一个很大的挑战。另外,如果用户允许引入新的流程,这将带来一个风险:相似的流程在公司的不同部门多次被创建,这样产生一些死的(不被使用的)模型。因此,我们提出**需求 6**:一个面向主题的演化业务信息系统必须为终端用户提供友好的支持模型管理的功能集合,比如:锁定、分支和模型合并,或者发现已有模型或带有某个任务的模型部件的语义搜索。

演化流程的一个目标是流程的持续改进以达到更高的业务性能。对流程模型结构化的改变(对一个或多个主题的内部行为或对通信模型)对业务性能而言要么是积极的要么是负面的,影响单个流程参与者(主题)或整个流程的输出。因此,让终端用户可以评估他们预定的改变所带来的影响是很重要的,比如在执行它们之前,通过仿真机制进行仿真,并获得关于执行改变的效果反馈以找到优化的潜力。这就引出了**需求 7**:一个面向主题的演化业务信息系统必须提供仿真工具,并提供关于在每个用户的上下文中流程性能的分析结果。

B. 讨论

将找到的软件需求进行分类,我们定义了为实现 S-BPM 驱动的演化业务信息系统中必须开发的三个领域:

- ◇ 模型管理(即访问控制、维护、语义模型检索)
- ◇ 通信和协同管理(公司层面的、部门内部的以及部门之间的)
- ◇ 持续流程优化的支持(验证&仿真、对所有利益相关者的流程性能反馈)

如果利益相关者能够对信息系统执行再设计,那么意味着在我们的上下文中他们必须对流程做出改变。但是单单建模是不足够的,这些变化应该可以 ad-hoc 方式完成而又不中断服务。例如:修改的模型必须在语法和功能正确性上被验证,新的流程版本可以被创建,或者在运行的流程实例的迁移必须被触发。在一般的 BPM 环境中,这些任务由建模专家和(或) IT 人员来完成。

因此,作为我们研究的初步结果,我们声明:将建模任务移动终端用户的同时也需要将其它的(技术的)任务移到终端用户。

需进一步的研究来澄清前面提到的任务哪些是要自动化的,这样它们能够被信息系统以向终端用户透明的方式执行。对于剩下的非自动化的任务,概念必须明确使得业务的利益相

关者们可以操作它们。

4 原型

在本节，我们介绍对于 S-BPM 驱动的业务信息系统的概念设计的初步结果，并且给出我们当前在进行两个模块原型的工作概览。

A. 概念设计

考虑到第 3 节中得出的软件需求，我们为 S-BPM 驱动的业务信息系统提出如下的概念设计，如图 3 所示。

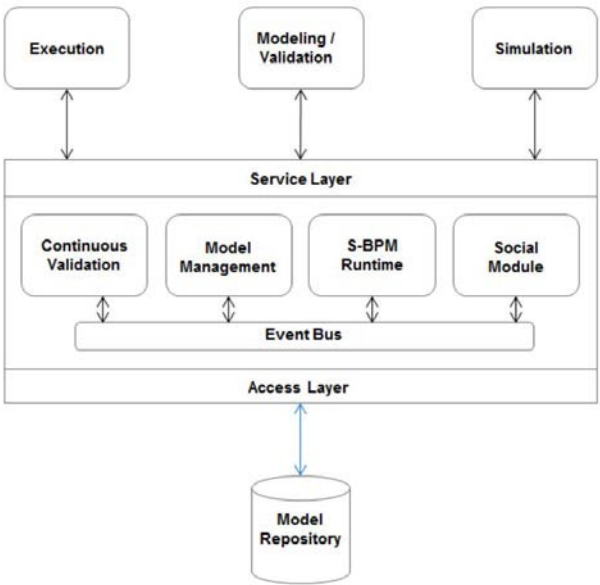


Figure 3. S-BPM 驱动的 EBIS 概念设计

由于利益相关者执行的变化需要被获得并且不需要中断服务就可以执行，和所有导出的软件需求都可直接或间接与模型操作关联，我们认为流程模型和终端用户应用的紧集成是可行的。为了这一目的，我们建议实现一个流程模型仓库作为一个中央的模型数据管理层，存储所有的应用模块。一方面，这允许模型的直接操作，使得它们有效地获得而不用上传到工作流引擎（这将引起服务中断）。另一方面，模型的修改，甚至是在实例级别的，可以立即转回到模型使得它们可以获得并被验证、分析和重用。另外，模型仓库应该提供分支、合并和流程模型的版本管理机制以满足需求 6。这样的话，我们有望将模型仓库作为改进模式开发和可变性管理的一个辅助。

模型仓库的访问可以通过一个访问层来进入，包括权限管理的功能需求（需求 1），特别是对授权的利益相关者限制模型部件。这层是为治理支持开发高级功能的前提。

应用的核心将是包含不同服务模块的服务层。

首先，S-BPM 运行时模块（工作流引擎）应该实现可以直接解释存储在仓库中流程模型，而在执行前不需要执行模型转换。我们假定这将开启额外的模型操作的可能性，比如：当一个用户执行 ad-hoc 偏离时，即时地移动到一个分支，这可用于促进模式开发。

第二，模型管理的模块的实现应该管理所有模型相关的操作，因此解决模式开发的特性。这包括自动的模型分支、合并，执行流程实例的模式迁移和模型搜索功能。另外，这一模块应该提供为重管理模型部件的服务和一个当 ad-hoc 变化产生时提供建议的推荐引擎（需求 5）。

第三，需要一个验证模块对模型来执行持续的验证操作。既然建模操作在任何时间都可以完成，我们认为一致性必须主动地被保证（需求 2）。

第四，社会模块为协作和向利益相关者的信息发布提供服务，这应该是被模型操作带来影响而触发出的（需求 4）。为了保证信息在需要的时候被传递，模块应该主动推送消息而不是等待消息请求。引入社会模块是实现反馈渠道的一部分。

这四个描述的服务模块不能被视为彼此独立的。例如：如果验证模块检测到问题，这个信息必须发布到订阅了该被影响的模型的利益相关者。另一个场景可能是在 ad-hoc 偏离的开始，模型管理模块应该给终端用户提供主动的模型路径推荐。为互相连接模块，我们建议增加通向服务层的事件总线。所有的模块应该监控这个总线并立即响应相关的事件，也将相关的事件推送给其它模块。

利益相关者通过不同的应用模块和系统交互。执行应用模块用于流程实例上的一般工作。建模和验证组件可使得利益相关者执行系统的再设计和验证其功能的正确性。仿真应用可使利益相关者测试模型变化带来的影响（需求 7）。

B. 当前工作

修改的 S-BPM 运行时模块

既然工作流引擎是必须和我们的系统一起工作，我们决定开始实现 A 节中提到的 S-BPM 运行时模块。为此，我们采纳已有的 S-BPM 工作流引擎并修改它，以至于它可以直接执行存储在流程模型库中的流程模型。

有了创建的原型，我们现在可以：

- ✧ 基于未改变的流程模型版本创建并执行流程实例；
- ✧ 在模型的主分支上执行改变，这将立即在执行环境中激活且不需要中断服务。

尽管我们在开始的案例分析中对可用性有了很好的经验，但几个问题还是有待研究，包括：

- ✧ 如何处理模型的不一致性？目前可以执行语法上不正确的模型，既然在将模型上传到运行时前，省略了以前执行的检查。
- ✧ 如何实现模型的自动化分支？目前，模型的变化将影响所有运行的实例。

动态模型开发

在探索的研究中，我们总结：修改的 S-BPM 运行时模块可以用于流程开发和验证。根问文献[4]，面向主题模型的验证可以通过执行角色扮演游戏来完成，其中每个利益相关者按定义在其建模的主题行为来做出动作。这个角色扮演游戏可以在有或者没有信息系统支持的情况下完成。

使用修改的 S-BPM 运行时模块，可以同时进行建模和执行，意味着模型改编可以立即执行应用和测试，而不要进一步的中断。因此，流程模型可以基于“建模-测试-建模”这样的循环迭代来创建。

为了评估我们的假设，我们正在结合建模和执行在一个集成的应用视图中进行原型实现，包括支持计算机辅助协同建模和角色扮演方法的社会组件的实现。

5 总结和未来研究

本文我们提出了关于使用 S-BPM 建立演化业务信息系统的初步研究结果。基于和 EBIS 定义的特性相比较显示：一般而言，S-BPM 方法是一个足够建立这样一个系统的方法。我们导出软件系统的需求，并且为 S-BPM 驱动的演化业务信息系统开发了一个概念设计。另外，我们描述了一个修改的 S-BPM 工作流引擎的原型，和一个集成建模和验证方法的原型，两者当前均处于需进一步的开发当中。

当实现这样一个系统是，我们已经从工作当中学习到一个关键点：需要将从建模专家最初的技术任务转移到终端用户。因此我们建议未来的研究工作如下：

- ✧ 建模专家的典型任务是什么？哪些需要转移到终端用户？

◇ 当转移这些任务到终端用户时，风险和挑战是什么？
◇ 复杂的技术任务（比如：流程模型的分支或合并）如何被业务用户所操作？
再一个重要的话题就是在建模流程时如何支持终端用户。为了创建合适的定性的模型，他们在如下方面需要及时的辅助：

- ◇ 相关的业务知识，比如：一致性的需求；
- ◇ 相关的建模知识，比如：最佳实践或反模式；
- ◇ 利用在模型中已经存在的解决方案来避免重复的工作。

因此，我们计划在下一步研究如何在面向主题的业务流程管理领域中充分利用好辅助的系统。

注：本文译自如下文献

Stephan Schiffner, Thomas Rothschädl, and Nils Meyer. "Towards a subject-oriented evolutionary business information system". Proceedings of 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW). IEEE, pp. 381-388, 2014.

2018/12/12

康国胜于复旦张江校区 SOARingLab 实验室
guoshengkang@gmail.com

6 参考文献

- [1] Alexander Nolte, and Michael Prilla. "Normal users cooperating on process models: Is it possible at all?". Proceedings of International Conference on Collaboration and Technology. Springer, pp. 57-72, 2012.
- [2] Alexander Nolte, and Michael Prilla. "Beyond Collaborative Model Usage and Development-A Model Lifecycle Approach for Lay User Modeling". Proceedings of MoRoCo@ ECSCW. pp. 49-56, 2013.
- [3] Michael Prilla, and Alexander Nolte. "Integrating ordinary users into process management: Towards implementing bottom-up, people-centric BPM". *Enterprise, Business-Process and Information Systems Modeling* Springer, pp. 182-194, 2012.
- [4] Albert Fleischmann, Werner Schmidt, Christian Stary, Stefan Obermeier, and Egon Börger. Subject-oriented business process management. Springer Science & Business Media, 2012.
- [5] Robert Gottanka, and Nils Meyer. "ModelAsYouGo:(re-) design of S-BPM process models during execution time". Proceedings of International Conference on Subject-Oriented Business Process Management. Springer, pp. 91-105, 2012.
- [6] Gustaf Neumann, Stefan Sobernig, and Michael Aram. "Evolutionäre betriebliche Informationssysteme". *Wirtschaftsinformatik*, Vol. 56, No. 1, pp. 41-47, 2014.
- [7] Joan Ernst Van Aken. "Design science and organization development interventions: Aligning business and humanistic values". *The Journal of Applied Behavioral Science*, Vol. 43, No. 1, pp. 67-88, 2007.
- [8] Matt Germonprez, Dirk Hovorka, and Uri Gal. "Secondary design: A case of behavioral design science research". *Journal of the Association for Information Systems*, Vol. 12, No. 10, pp. 662, 2011.
- [9] Matt Germonprez, Dirk Hovorka, and Fred Collopy. "A theory of tailorable technology design". *Journal of the Association for Information Systems*, Vol. 8, No. 6, pp. 21, 2007.