

PackML UNIT/MACHINE IMPLEMENTATION GUIDE

Part 1: PackML Interface State Manager

摘要

“PackML Interface State Manager” 文档是 PackMLUnit/Machine Implementation Guide 的第二版。PackML Unit/Machine Implementation Guide 是在 ANSI/ISA TR88.00.02-2022 的基础上编写的最佳做法建议。本指南的宗旨是帮助知识水平各异的企业在 Unit/Machine 上实施 PackML interface。本文档介绍的最佳做法建议源于若干 OMAC 成员企业及其他用户实施 PackML interface 的真实案例。未来，在制造环境和供应链中数据将发挥巨大价值，而 PackML interface 是积累协调数据以实现制造环境数字化并获得工业 4.0 应用的关键之一。使用 PackML interface 的好处之一是标准化的数据模型以及对 Unit/Machine 的 State 和 Mode 的共同理解。它将提供根据上级 IT 系统（如 SCADA 和 MES）协调数据的可能性。在本地部署的系统上使用 EDGE 和云系统还将增加从许多 unit/machine 获取 PackML 数据并将数据发送到有利于使用数据的 IT 系统的可能性。

ISA TR88.00.02-2015 和新版 ISA TR88.00.02-2022 之间的主要变化是，新版本更符合核心标准 ISA-88.00.01，并通过配方执行订单（即批次）。这些变化包括更新 State Model、删除“远程”界面，并将通过“配方”描述 Unit/Machine 配置的 PackTag 添加到监控系统和堆栈灯信息。

本文档是指导企业打造 PackML Unit/Machine 系列文档的第一部分。后续的文档将陆续推出，并侧重于 PackML Machine Software Code Structure、PackML Network Connections、PackML and Safety、PackML and Line Integration 以及 PackML User Interface - HMI 等主题。

Author: Ph.D. Carsten Nøkleby, SESAM-World

Translated: Mitsubishi Electric Corporation

1. 目录

2. 执行摘要	4
2.1 宗旨	4
2.2 目标	4
2.3 背景	5
2.4 编写中的相关 PackML 文档	5
3. 本文档的对象范围	6
3.1 PackML Interface State Manager	8
4. 术语、定义和缩写	9
4.1 ISA 88.01	9
4.2 ISA 88 物理模型	9
4.3 ISA 88 Recipe Management – Unit/Machine 控制配方参数	10
5. 识别和定义单元	13
6. PackML Interface State Model	16
6.1 PackML State Model 的语法	16
6.2 PackML Interface State Model	17
6.3 PackML State Model 中使用的各种颜色	18
6.4 从 Unit/Machine 角度看待 PackML State Model	19
6.5 从内部和外部角度看待 PackML State Model	20
6.6 从 Unit/Machine 参数角度看待 PackML State Model – 生产订单	21
6.7 PackML State Model 和堆栈灯	24
7. PackML Event State Manager	26
7.1 Resetting	27
7.2 Starting	29
7.3 Execute	31
7.4 Holding	33
7.5 Unholding	37
7.6 Suspending	41
7.7 Unsuspending	44
7.8 Completing	47
7.9 Stopping	49
7.10 Aborting	51
7.11 Mapping table	54
7.12 Mapping Alarms to event triggers to PackML	56
8. PackML Control Commanddefinitions	57

8.1	PackML Commands.....	57
8.2	Unit HMI 上和来自外部系统的命令的规范.....	60
9.	PackML Interface State 定义	61
10.	Unit Control modes.....	65
11.	PackTag	68
11.1	PackTag 说明和定义	72
11.2	PackTag 2015 到 PackTag 2022 的变迁	72
12.	Unit Controller Functionality 示例.....	77
12.1	Syntax 和 Symbolic 说明	77
12.2	准备 Unit/Machine	78
12.3	启动 Unit/Machine	79
12.4	从 Unit/Machine Panel/HMI 停止 Unit/Machine	81
12.5	从外部系统停止 Unit/Machine	82
12.6	操作员中断和暂停使用 Unit/Machine	84
12.7	中断后从外部系统恢复 Unit/Machine	85
12.8	在 Unit panel/HMI 停止 Unit	86
12.9	Abort Unit/Machine	88
12.10	在 Stop 或 Abort 后从 Unit Panel/HMI 重新启动 Unit.....	89
12.11	在 Stop 或 Abort 后从外部系统重新启动 Unit.....	90
12.12	Unit 的报警和警告	91
12.12.1	PackML alarm 和 event state management table	92
12.13	背景任务: OEE.....	94
12.14	背景任务: Mode 和 State	95
12.15	Suspend: 缺料或饱和	97
12.16	Unsuspend	98
13.	参考信息	99
13.1	定义和缩写.....	99
13.2	参考资料	100
13.3	参与编写本指南的人员名单	101
13.4	Support	103
13.5	The author	103
14.	Version History	104

本页最初留空。

2. 执行摘要

2.1 宗旨

PackML Interface State Manager 文档是在 ANSI/ISA TR88.00.02-2022 的基础上编写的最佳做法建议 PackML Unit/Machine Implementation Guide 的第一部分。本指南将帮助知识和经验水平各异的企业熟悉 ANSI/ISA TR88.00.02-2022 技术报告以及了解如何将本指南中所述的步骤和流程运用到现实中的机器应用。PackML Interface State Manager 侧重于在机器上执行生产订单，而不是机器如何执行其控制功能。Unit/Machine 的定义是执行一项或多项重要处理功能的一系列物理设备和控制功能。Unit/Machine 可以是一台机器，也可以是整个包装生产线的某个组成部分。

PackML Unit/Machine Implementation Guide 第一版的背景信息源自 2015 年、2016 年召开的工作组会议和 ISA TR88.00.02-2015 版本。此外，本文也融入了工作组成员的实践经验，以便构建即刻可用、通用且独立于供应商的实施模型。

第二版的背景信息是更新的 ISA TR88.00.02-2022。

与 PackML Unit/Machine Implementation Guide 第一版的主要变更之处是，更新并扩展了 PackML State Model，增加了 PackML Command 的范围。最大的变化与 PackTag 有关，其中添加了处理生产订单数据的配方结构，以及堆栈灯的标签，并删除了远程接口。

这本 PackML Unit/Machine Implementation Guide 适用于机器供应商、系统集成商和最终用户。本指南介绍了 Unit/Machine 要想符合 ANSI/ISA TR88.00.02-2022 所述的 PackML 标准所应该满足的要求，并提供了可作为特定机器指南的实施示例。

2.2 目标

数据将成为新商业模式和工作方式的驱动力。不久的将来，在制造环境和供应链中数据将发挥巨大价值，而 PackML interface 是积累协调数据以实现制造环境数字化并获得工业 4.0 应用的关键之一。使用 PackML interface 的好处之一是标准化的数据模型以及对 Unit/Machine 的 State 和 Mode 的共同理解。它将提供根据上级 IT 系统（如 SCADA 和 MES）协调数据的可能性。在本地部署使用 EDGE 和云还将增加从许多 unit/machine 获取 PackML 数据并将数据发送到有利于使用数据的 IT 系统的可能性。

这本 PackML Unit/Machine Implementation Guide 提供了 PackML interface 要求的定义。例如，当设备级别发生错误或事件并生成报警或警告时，如何处理不同条件的实际示例。它还解决了协调各种用户不同的解读。本指南提供了如何根据最终用户的特定要求配置 PackML State Model 转换的示例。

此外，本指南还考虑了机器供应商的成熟度水平，以及机器供应商部分或全部实施 PackML 的准备程度或能力。

最终用户和机器供应商都可以获得以下好处：

- 轻松集成 Unit/Machine 和测试接口
- 降低整条包装生产线的集成成本。
- 与监控系统使用相同的操作员接口
- 更快的接口规范
- 降低商业合同中的风险/不确定性

- 减少培训工作
- 可靠的数据（例如 OEE、能源数据）
- 更换生产线上的一个 Unit/Machine
- 车间机器和设备的标准化语义模型
- 能够数字化制造环境和供应链的标准
- PackML 结合 Edge 技术和本地部署的云系统将加快迈向工业 4.0 的步伐

2.3 背景

OMAC 最终用户希望通过统一的接口与车间中的 unit/machine 交互。提供该统一接口将确保与监控控制系统轻松集成。届时，所有单元都使用类似的接口，而且这些单元都具有相同的数据结构接口。

根据 PackML State Model 应用统一的用户接口后，操作员即便在面临计划外停机时也能妥善应对。此外，所有单元都可使用通用的事件处理机制，有利于最终用户采用适用于所有单元的方式查明停机原因。

基于 PackML 和 ISA TR88.00.02-2022 的统一数据接口将有助于单个 Unit/Machine 的数据集成和结构化，并提供从监控控制系统实现通用接口的可能性，例如 Manufacturing Execution System (MES) 或 Supervisory Control and Data Acquisition (SCADA)。

本文所述的接口基于 International Society of Automation (ISA) 定义的国际参考模型 PackML 所构建。该接口遵循 ANSI/ISA TR88.00.02-2022 中所述的 PackML State Model。

建议了解 ISA 88.00.01 批量控制标准，它是 ISA 88 定义域相关标准和 TR88.00.02-2022 技术报告的基础。该技术报告是对 ISA 88.00.01 关于离散机器和制造环境的解释。阅读这本 PackML Unit/Machine Implementation Guide 之前必须对 TR88.00.02 技术报告有着基本了解和认识。事实上，我们强烈建议在阅读本文时将该报告放在手边，以便随时参考。

Unit/Machine 的定义是执行一项或多项重要处理功能的一系列物理设备和控制功能。Unit/Machine 可以是一台机器，也可以是整个包装生产线的某个组成部分。本文档中，unit 和 machine 两词可以互换。

机器供应商的定义是提供兼容 PackML 的 Unit/Machine 的公司或组织。最终用户的定义是在生产设施中使用兼容 PackML 的 Unit/Machine 的公司或组织。

2.4 编写中的相关 PACKML 文档

本文档是指导企业打造 PackML Unit/Machine 系列文档的第一部分。更多的文档¹将陆续推出，并侧重于以下主题：

- Part 1: PackML Interface State Manager (可从 OMAC 获得)
- Part 2: PackML Standard Representation of PackTags in an OPC UA server (可从 OMAC 获得)
- Part 3: PackML Network Connections
- Part 4: PackML and Safety
- Part 5: PackML and Line Integration
- Part 6: PackML User Interface - HMI (可从 OMAC 获得)

¹列表内容不完整，文档标题可能会有所更改。Part 1、Part 2 和 Part 6 可从 OMAC 主页获得。

3. 本文档的对象范围

PackML Unit/Machine Implementation Guide 第一部分仅侧重于作为个体的 Unit/Machine 及其 PackML Interface 的实施，而与其他 Unit/Machine 或设备无关。

本文并不涉及若干台 Unit/Machine 在生产线上是如何工作的内容，这将包括在 OMAC Part 5: PackML and Line Integration。

PackML 规范是 ANSI/ISA-TR88.00.02-2022、机器和 Unit/Machine State 的实施。作为 ANSI/ISA-88.00.01 的实施示例，它涵盖以下两个应用示例：

1. PackML Interface State Manager²
2. PackML Machine State Manager（不在本文介绍范围内³）

PackML Unit/Machine Implementation Guide 第一部分的侧重点是“PackML Interface State Manager”，也就是图 1 中的上方（蓝色）部分。

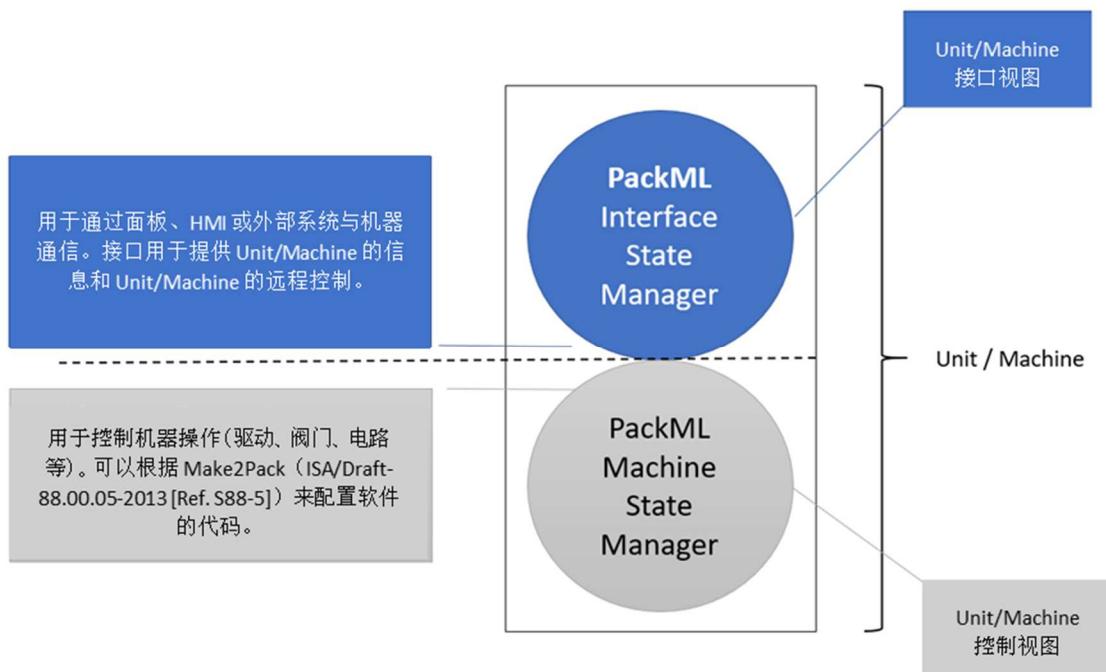


图 1：PackML Interface State Manager 和 PackML Machine State Manager 的定义

想要将 PackML State Model 用于其内部 Machine State Manager 的机器供应商有两种情况，如图 2 所示：

- 1) PackML Gateway Unit/Machine
- 2) Full PackML compliant Unit/Machine

²ISA-TR88.00.02 介绍了一种基本状态模型，在此模型中，可以使用 PackTag 数据区域并利用 PackTag Command、Status 和 Admin tag，将动作驱动到 OEM 的“PackML Machine State Manager”中。

³完整的 Make2Pack 分解方案详见 ISA88.00.05-2013-Draft [Ref. No. S88-5]。

PackML Unit/Machine Implementation Guide 第一部分的重点是 PackML Gateway 解决方案，并涵盖了 PackML Interface State Manager。

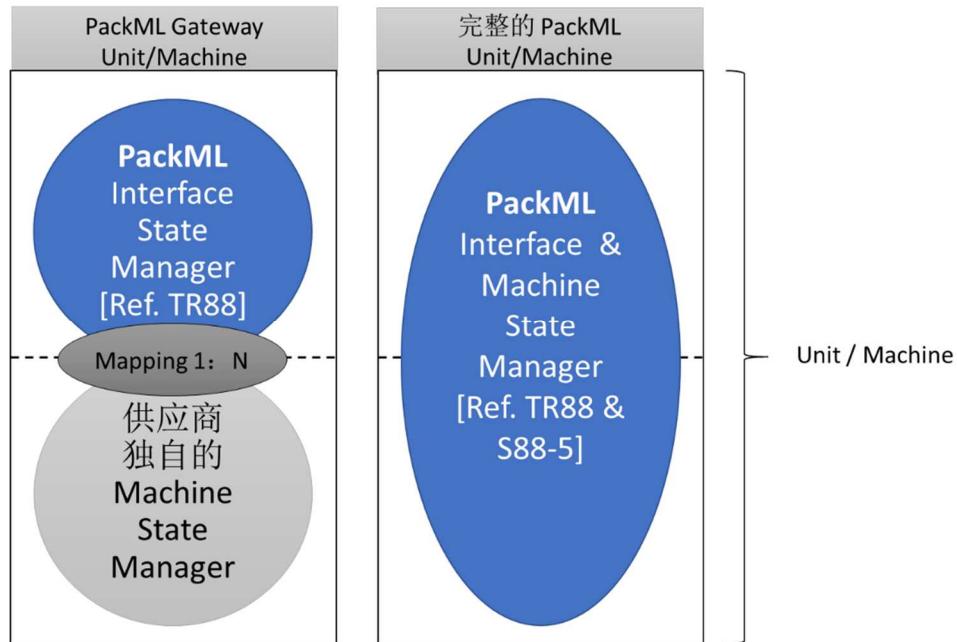


图 2: PackML Interface State Manger 和集成 Machine State Manager & PackML Interface 之间的映射关系⁴

要实施 PackML Gateway，机器供应商必须将 PackML Interface State Model 映射到其特定的内部状态逻辑³。

拥有 PackML Gateway Unit/Machine 的机器供应商需要在 PackML Interface State Manager 与其特定的 Machine State Manager 之间建立起映射关系。通常来说，所有机器供应商都应该能映射到 PackML Interface State Manager。

当机器供应商拥有基于 PackML 打造的 Machine State Manager，则无需在 PackML Interface State Manager 与 Machine State Manager 之间建立映射，因为它们是相同的⁵。

PackML Gateway 方案适合应用于旧设备以及对内部机器状态逻辑有着固有要求的机器供应商。

要获得完整的 PackML Unit/Machine，必须根据 ISA-88 [Ref. S88-1]、PackML [Ref TR88] 和 Make2Pack [Ref. S88-5-Draft] 对该 Unit/Machine 重新编程。

⁴ 该映射表见第 7.11 节。

⁵ 在一台机器有多个 State Manager 的情况下，应将每个 State Manager 呈现给外部系统，并定义适当的映射。例如，托盘堆垛机处理不同包装模式的托盘时，可能会为处理单个托盘位置的机器的每个部分配备 PackML State Manager。

3.1 PACKML INTERFACE STATE MANAGER

PackML Interface State Manager 用于通过面板、HMI 或外部系统与 Unit/Machine 进行通信。该接口用于获取 Unit/Machine 的状态、启动 Unit/Machine、停止 Unit/Machine、更改配方参数等。

PackML Interface State Manager 在 HMI 或其他外部控制系统（如其他 Unit/Machine 或监控控制系统）与 Unit/Machine 之间提供了单一通信接口。

PackML Interface State Manager 可以与 Machine State Manager 在同一个控制系统 (CPU) 中实施，也可以实施在单独的控制系统中。一个完整的 PackML Unit/Machine 将具有一个通用接口和一个完全与 Machine State Manager 集成的 Machine State Manager。请参见图 2: PackML Interface State Manager 和集成 Machine State Manager & PackML Interface 之间的映射关系。

从通信角度来说，一个 Unit/Machine 具有单个 PackML Interface State Manager，后者将提供一个接口处理来自 HMI 和/或外部控制系统的 Unit/Machine 操作。

例如，一个 Unit/Machine 可以有三个内部驱动控制器，但从用户和操作员角度来看，它会被视为一个 Unit/Machine。又例如，由若干进料传送带、一个机械臂和若干出料传送带构成的托盘拆垛机可以被视为一个 Unit/Machine。单个 Unit/Machine 的定义通常与执行的主要处理范围有关。

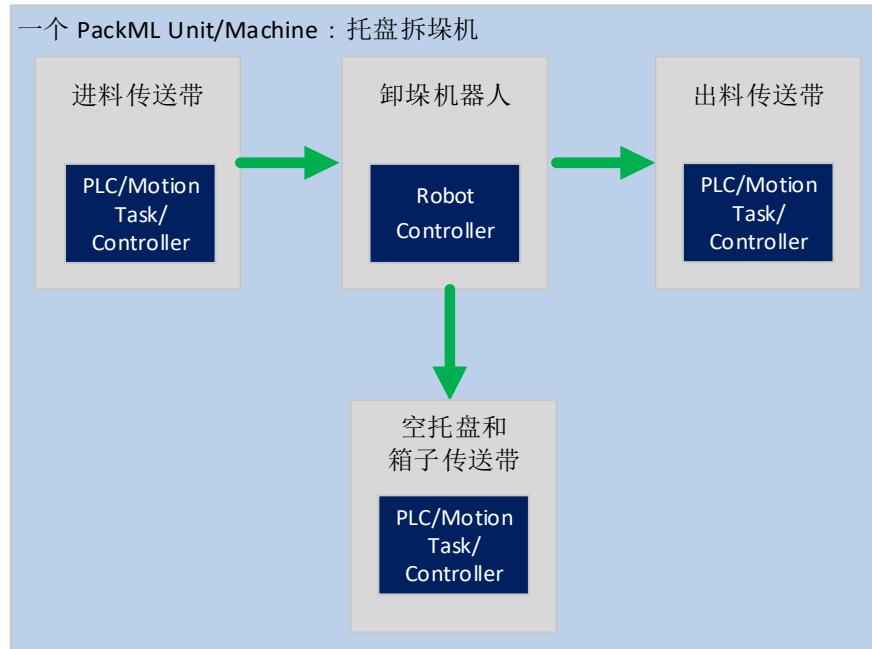


Figure 3: 由多个控制器组成的托盘拆垛机示例。

4. 术语、定义和缩写

4.1 ISA 88.01

ANSI/ISA-88.00.01-2010 [Ref. S88-1] 标准并不仅限于批次生产行业！本标准适用于不同类型的生产过程：离散、连续和批次生产行业。

S88-1 是一种用于设计和操作用于柔性机器和生产控制系统的模型和方法。它的架构和模型独立于底层控制系统（PLC、DCS、PC 等）以及底层基本控制算法。

S88-1 的目标是在控制系统工程师不介入以及无需对控制系统编程的情况下也能开发配方。该元素被纳入 ISA TR88.00.02-2022，这是对 S88-1 与离散制造机器和设备的新解释。

4.2 ISA 88 物理模型

ANSI/ISA-88.00.01-2010 [Ref. S88-1] 和 ANSI/ISA-TR88.00.02-2022 [TR88] 通过物理模型定义设备和机器的层级，如图 4 所示。PackML Machine Implementation Guide 仅侧重于下述物理模型中的 Unit/Machine。ISA 88 物理模型中的 Unit/Machine 相当于托盘拆垛机、装袋机、灌装机、封盖机、装箱机、托盘堆垛机等 machine。

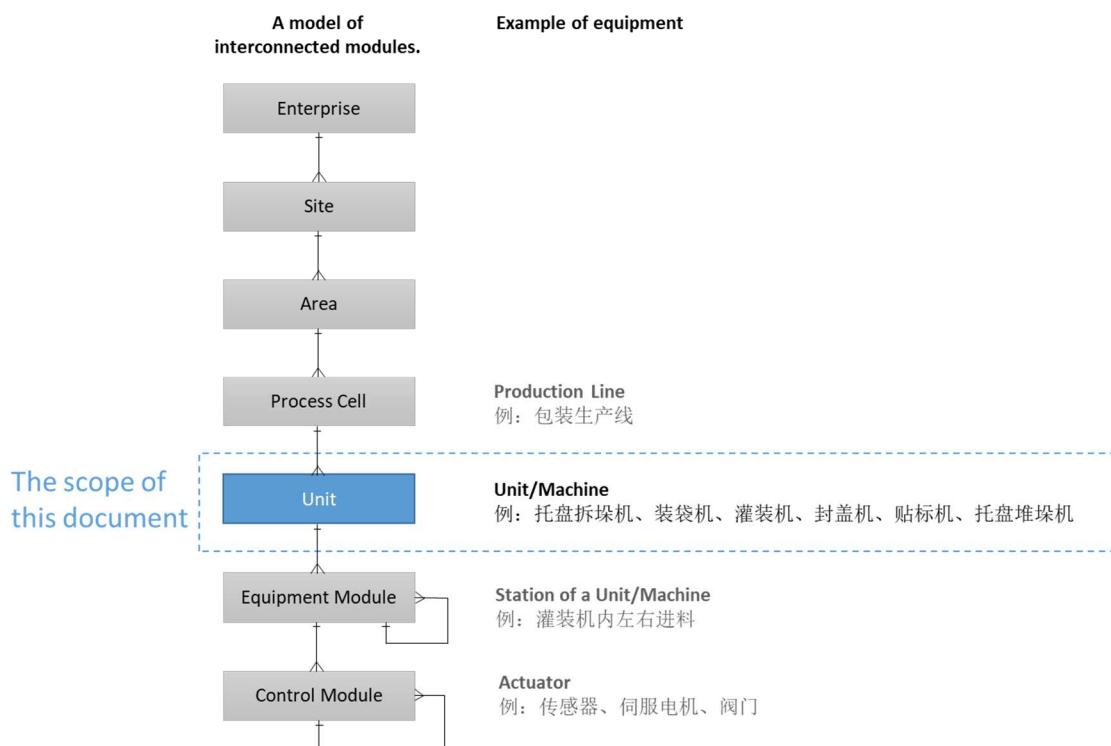


图 4：ISA 88 物理模型与包装设备的映射关系

ISA 物理模型具有以下层级：

- Enterprise / Site / Area（超出本文范围）。
- Process Cell
 - 它涉及从开始执行单个配方到完成生产订单所需的一整套设备。
 - 由一个或多个单元构成
- Unit/Machine - Machine
 - 一套设备模块和控制模块
 - 通常以大型设备为中心
 - 通常处理或包含完整的生产订单。
 - 可能只处理或包含完整生产订单的一部分。
 - 一次无法处理或包含一个以上的生产订单⁶
- Equipment Module
 - 一套控制模块或其他设备模块
 - 可以执行有限数量的次要处理活动（即阶段）
 - 包含所有必要的处理设备来开展这些处理活动
 - 例如，一个设备模块可能由循环泵、冷水阀、蒸汽阀和温度控制器构成。此时，该设备模块代表一个温度控制系统
- Control Module
 - 一套传感器、致动器、其他控制模块和相关的处理设备
 - 从控制角度看，起着单个实体的作用
 - 通过其传感器和致动器直接连接到工序
 - 控制模块示例：阀门、泵、马达、压力控制器、限位开关等

4.3 ISA 88 RECIPE MANAGEMENT – UNIT/MACHINE 控制配方参数

ISA-TR88.00.02-2022 进行了更改，更符合 ISA-88.00.01 基本标准，该标准定义了一个物理模型、配方模型和程序模型，使物理系统在生产产品方面具有灵活性。该产品以配方为基础，配方中包含有关工艺变量（温度、速度、扭矩等）和成分（纸板、胶水、铝箔、托盘等）的信息。这是标准的一大变化，因为它是未来实现柔性生产系统的关键。

与工序有关的 Unit/Machine 参数用于指定 Unit/Machine 需要执行哪个任务。与工艺相关的 Unit/Machine 参数表示 Unit/Machine 应为特定生产订单执行的操作，例如胶水量或打印机标签格式。相对于 ISA 88 [Ref. S88-1] 而言，此类参数被称为控制配方参数。

⁶ 如果需要一次运行多个生产订单，则需要在机器上部署若干个 PackML Interface State Manager。另一种方案是设置缓冲区，在缓冲区中缓冲要执行的若干生产订单。随后，机器将根据 PackML Interface State Manager 逐步完成生产订单。一个物理 Unit/Machine 作为托盘堆垛机的例子，它能够一次运行两个生产订单，因此处理每个生产订单的设备的每个部分都需要一个 PackML Interface State Manager。

本指南侧重介绍控制配方参数，包括过程变量和成分。范围内没有其他配方管理元素。

控制配方参数是指与特定工序有关的参数，这些参数可以在 Unit/Machine 上配置以执行特定任务。例如，托盘堆垛机可能具有指定包装形状和托盘层数的参数。

PackML Unit/Machine 必须能处理由控制配方参数表示的工序相关机器参数，即与工序密切相关且有关的参数。在 ISA TR88.00.02-2022 中，配方参数由 PackTags Command.Recipe[#] 和 Status.Recipe[#] 表示。

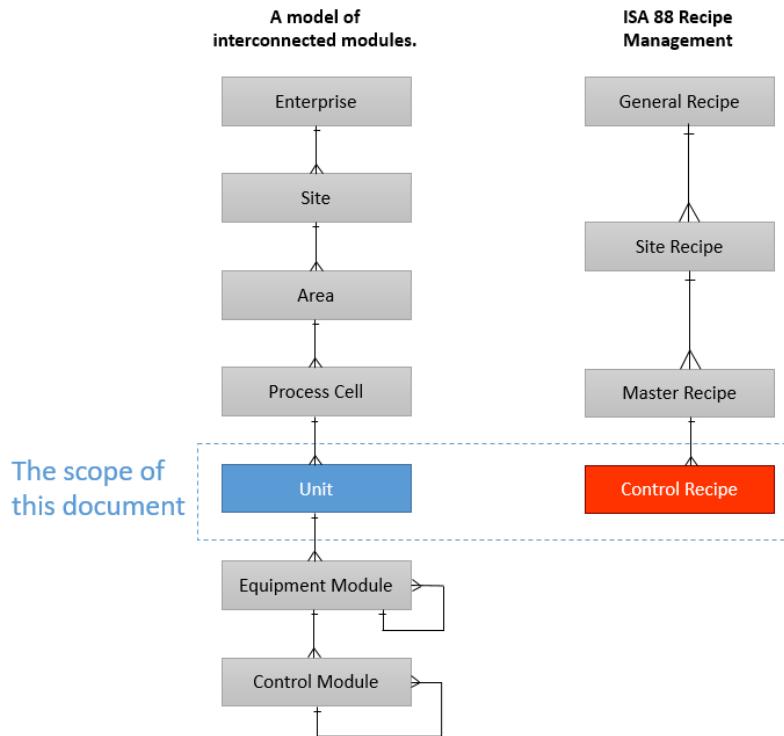


图 5：ISA 物理模型和 ISA 配方管理

配方是指不可或缺的一组信息，用来唯一定义特定产品的生产要求。配方和设备之间的关系如图 5 所示。

- **General Recipe:** 每个生产品都有一个通用配方，它们在企业级别维护（在 ERP 系统中维护，并在物料清单 - BOM 中定义）。
- **Site Recipe:** 每个生产点和生产品都有一个生产点配方，它们在生产点维护，用于当地物料、语言或生产段。
- **Master Recipe:** 每条生产线（Process Cell）和生产品都有一个主配方。
- **Control Recipe:** 每个生产订单都有一个控制配方，用于描述生产特定产品时需要在 Unit/Machine 中设置的参数。举例来说，托盘堆垛机 Unit/Machine 接收生产订单的三个与工序有关的参数：即包装形状、层数和层间距。

控制配方添加到 ISA-TR88.00.02-2022 版本中，包含过程变量和参数。此外，配方编号已添加到 Command Tag 中，以指示在 Unit/Machine 上执行的配方。

提示：Unit/Machine 的控制配方参数需要与机器供应商和最终用户共同定义。工序相关的参数不必反映 Unit/Machine 内部参数。Unit/Machine 可能具有成百上千个内部参数，但最终用户不会对这些参数感兴趣。只有参数定义了特定产品的 Unit/Machine 主要配置。

提示：一组控制配方参数与生产订单具有 1:1 的映射关系。在某些行业，一个生产订单可能称为一个“生产批次”，或直接称为批次。

提示：从 Unit/Machine 的角度来说，这并没有什么不同，不论控制配方参数是从其他系统传输而来的，还是在 Unit/Machine 本身的 HMI 接口上直接输入的。

提示：确定控制配方参数的范围时，以下几点很有用，也可以说至关重要：

- 对工序来说，它们是与质量密切相关的参数⁷
- 开展风险评估，确定哪些参数对生产的影响更大

举例来说，热封合应该具有下列与工序有关的 Unit/Machine 控制配方参数：速度、时间温度、压力、接触区域、材料温度、箔片厚度变化、材料层。

如果工序相关的 Unit/Machine 参数密切相关且对 Unit/Machine 至关重要，则可以联合最终用户的专家，通过开展故障模式及后果分析 (Failure Mode and Effects Analysis, FMEA) 加以确定。

提示：控制配方的目的是能够重新编程或配置 Unit/Machine，使最终用户能够灵活地引入新产品或设置。

⁷ 例如，质量相关的参数可能包括配方名称/编号以及其他产品指定参数（Unit/Machine 有一个本地配方管理系统，可自动将下载的配方名称/编号所需的所有内部机器参数集加载到机器中）。

5. 识别和定义单元

Unit/Machine 的定义是执行一项或多项重要处理功能的一系列物理设备和控制功能。Unit/Machine 可以是一台机器，也可以是整个包装生产线的某个组成部分。

Unit/Machine 会以通用 Unit/Machine 接口从功能或物理上进行定义。PackML Interface State Manager 在 HMI 或其他外部控制系统与 Unit/Machine 之间提供了单一通信接口，如图 7 所示。

Unit/Machine 上出现错误可能会导致该 Unit/Machine 内的所有子系统停机，并生成报警或警告。通常来说，安全回路决定了包装生产线上 Unit/Machine 的范围。

如果 Unit/Machine 上的错误只导致部分子系统停机并生成报警或警告，则可将各个子系统定义为单个 Unit，但并非一定要如此不可。

提示：PackML Interface 是为处理特定产品的单元的每个部分定义的。一个 Unit / Machine 一次只能处理一个生产订单。

例如，某个 Unit/Machine 具有三个内部驱动控制器，但从用户和操作员角度来看，它会被视为一个 Unit/Machine。

又例如，由进料传送带、机械臂和出料传送带构成的托盘拆垛机可以被视为一个 Unit/Machine，如图 6 所示。

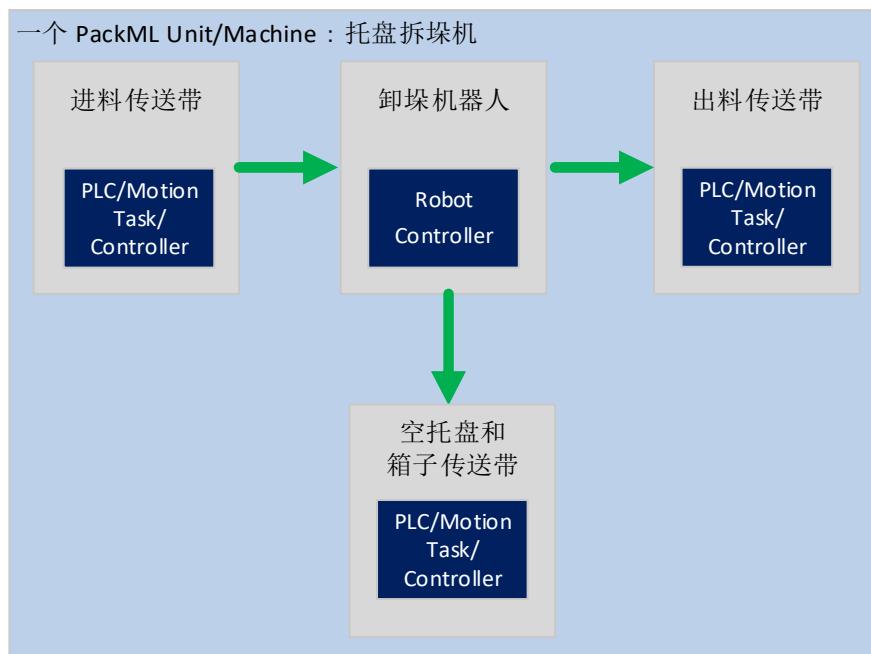


图 6：由不同系统要素组成的 PackML Unit/Machine

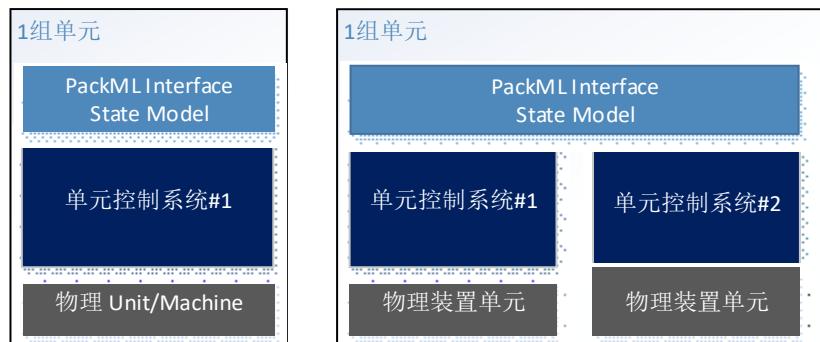


图 7：映射到一个 PackML interface 的 Unit/Machine 示例，即使 Unit/Machine 包含多个控制系统。

对于执行多个独立工序的 Unit/Machine，每个独立工序都需要一个 PackML Interface。例如，一个托盘堆垛机在同一个生产线上具有两个独立的包装单元，因此需要两个 PackML Interface。

Unit/Machine 可通过两种不同方法实施 PackML State Manager Gateway (PackML GW)；一种是将 PackML Gateway 的软件代码设为 Unit/Machine 的一部分，另一种是将 PackML Gateway 代码放在外部系统（如 PLC 或微控制器）中定位和执行。

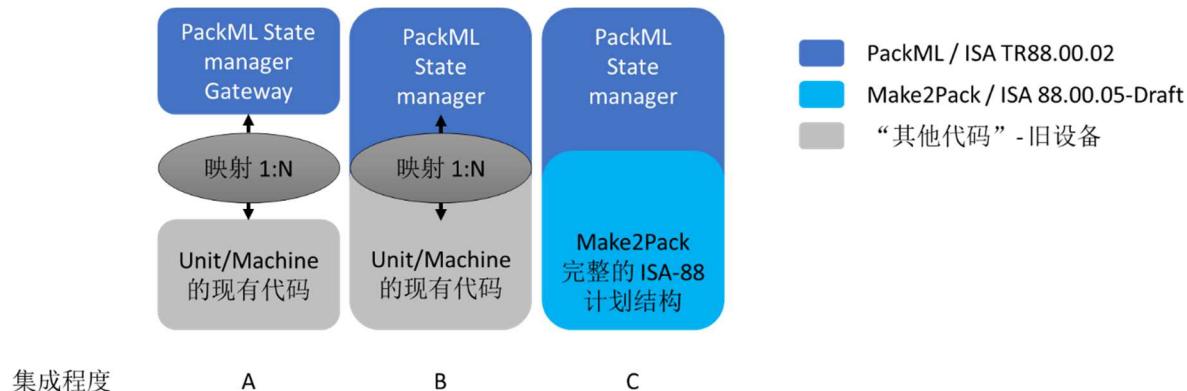


Figure 8: PackML implementation 水平

PackML integration 水平主要有三种类型，如 Figure 8 所示：

- **Unit/Machine 集成程度类型 A:**
Unit/Machine 使用在本地或外部运行的 PackML Gateway – 此情形主要表示 PackTag⁸。
Unit/Machine 由现有 Machine State Model 控制，而内部 machine State 通过 PackML Gateway 映射到 PackML Interface State Model。这意味着，机器供应商需要定义机器现有代码与 PackML Gateway 之间的映射关系。PackML Gateway 将使用 PackML interface 和 PackTag 与其他系统通信。

⁸ 请参见本文第 0 节了解 PackTag 的介绍。

- **Unit/Machine 集成程度类型 B:**

Unit/Machine 将 PackML Gateway 代码和 PackML Interface State Manager 实施在 Unit/Machine 控制系统的软件代码中。集成程度 A 和 B 之间的主要区别是 PackML Gateway 代码的位置不同。此实施方案代表 PackTag 和 PackML State Model。PackML interface 将映射到机器的现有控制代码。

- **Unit/Machine 集成程度类型 C⁹:**

Unit/Machine 程序的结构根据 PackML 和 ISA 88.00.05-Draft 编程规则划分。有关详细信息，请参见 Make2Pack [Ref. No. S88-5]。Unit/Machine 将使用 PackML interface PackTag 与其他系统通信。PackML interface 的完整实施与 Unit/Machine 程序代码集成。

⁹ 需要注意的是，集成程度 C 有可能支持（也有可能不支持）全面使用 Make2Pack (M2P)，具体取决于技术提供商提供的特定模板。另外，技术提供商可以为他们的 OMAC PackML/TR88 模板提供两个模板，一个支持 Make2Pack，一个不支持 Make2Pack。

6. PACKML INTERFACE STATE MODEL

6.1 PACKML STATE MODEL 的语法

PackML Interface State Model 基于两个主要元素：

- Commands
 - o 将 Unit/Machine 从一种状态切换到另一种状态的触发器（例如，Unit/Machine 上的按钮或通过网络发送的外部命令）
- States
 - o Acting states (Unit/Machine 执行某个动作时所处的状态)
 - o Waiting states (Unit/Machine 的稳定状态) 等待状态需要一个命令才能进入下一个状态。

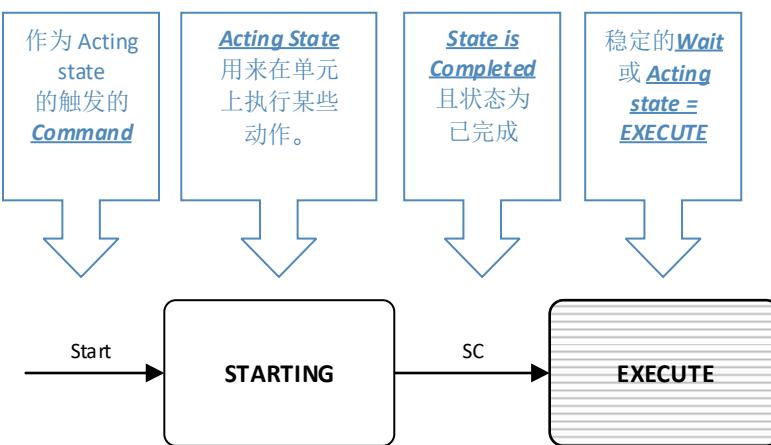


Figure 9: PackML State Model 的语法

6.2 PACKML INTERFACE STATE MODEL

ISA-TR88.00.02-2022 中的 PackML Interface State Model 更符合与状态变化相关的物理 Unit/Machine 的处理。

PackML Interface State Model 是一种状态模型，它通过标准化的方式表示 Unit/Machine 状态。接口说明基于状态模型、状态说明和相关的控制命令。图 10 为接口状态图 (ref. ISA TR88.00.02-2022)。

提示：建议在生产模式中实施全部 17 个 PackML state。但是，机器供应商和最终用户可协商 17 state 外要实施的其他状态。

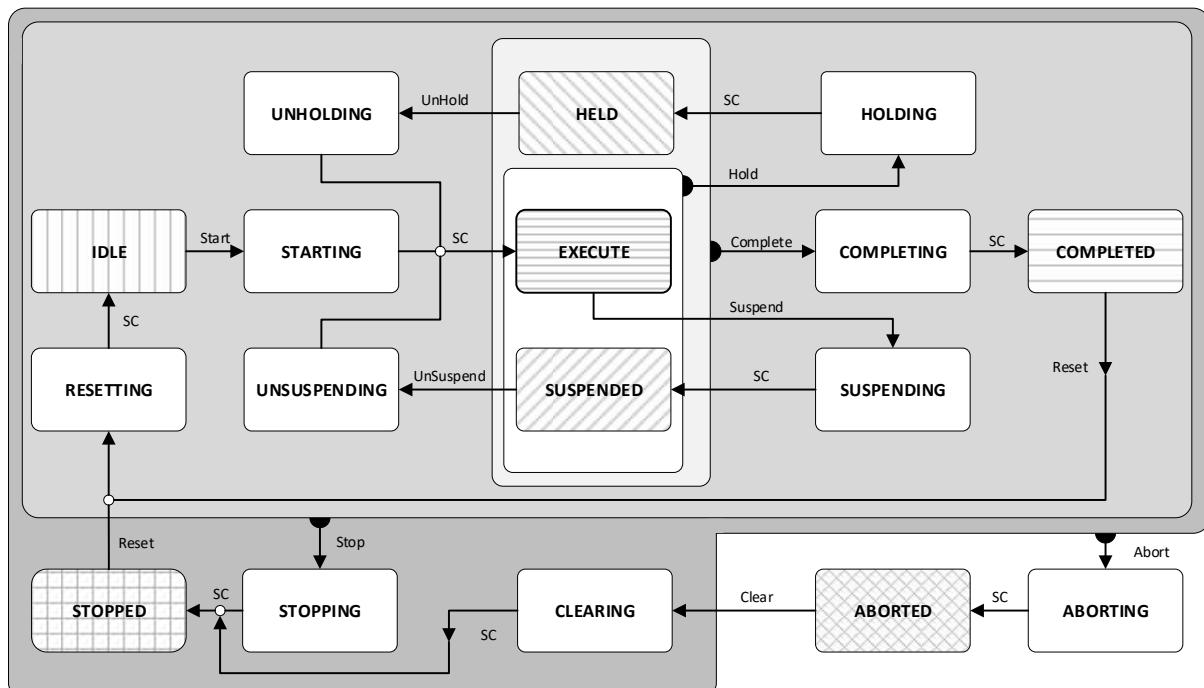


图 10: PackML Interface State Model

表 1: 不同图案表示不同状态的 Interface State Model

图案	状态	说明
	EXECUTE	Acting State - Unit/Machine 处于稳定的 acting state - Unit/Machine 正在生产。
	STOPPED IDLE COMPLETE	Wait State – 一种稳定的状态，用来说明 Unit/Machine 已完成一组指定条件。在此状态下，Unit/Machine 保持某个状态，直至一个命令使其切换到 Acting state。Unit/Machine 通电，但无动作。

图案	状态	说明
	RESETTING STARTING SUSPENDING UNSUSPENDING COMPLETING HOLDING UNHOLDING ABORTING CLEARING STOPPING	Acting State – 该状态表示某些处理活动，例如逐渐加速。它意味着在一段有限时间内或在达到特定条件前，按照逻辑顺序执行某些处理工序一次或反复执行这些处理工序，例如在 Starting 状态下，先检查收到的数据的质量和有效性，然后再提速执行相关操作。
	HELD ABORTED	Wait state – 该状态表示单元上的错误状态，生成报警或警告。在此状态下，Unit/Machine 未在生产，直至操作员切换到 EXECUTING 状态。该状态会将 Unit/Machine 操作挂起，以便在此期间清除物料堵塞或安全地更正设备故障，然后才会恢复生产。
	SUSPENDED	Wait State – 在此状态下，Unit/Machine 不生产任何产品。它要么停止运行，要么继续空转，直至外部工序条件恢复正常。外部工序条件恢复正常后，它将从 SUSPENDED 状态切换到 UNSUSPENDING 状态，在此期间通常无需操作员介入。

6.3 PACKML STATE MODEL 中使用的各种颜色

PackML Interface Model 中有多种颜色来表示不同的状态，会将所有颜色指南在另一本手册：
PackML User Interface – HMI 中介绍。

第 6.7 节 *PackML State Model* 和堆栈灯中提供了 ISA TR88.00.02-2022 的堆栈灯示例。关于用户接口和 HMI 的其他详细信息，请参见 PackML Guideline: *Part 6: PackML User Interface – HMI*。

6.4 从 UNIT/MACHINE 角度看待 PACKML STATE MODEL

下图显示了操作员可使用 HMI 按钮进行干预¹⁰:



四个按钮已添加到状态模型中，如图 11 所示，来协助说明 Unit/Machine HMI 与 PackML State Model 的关系。

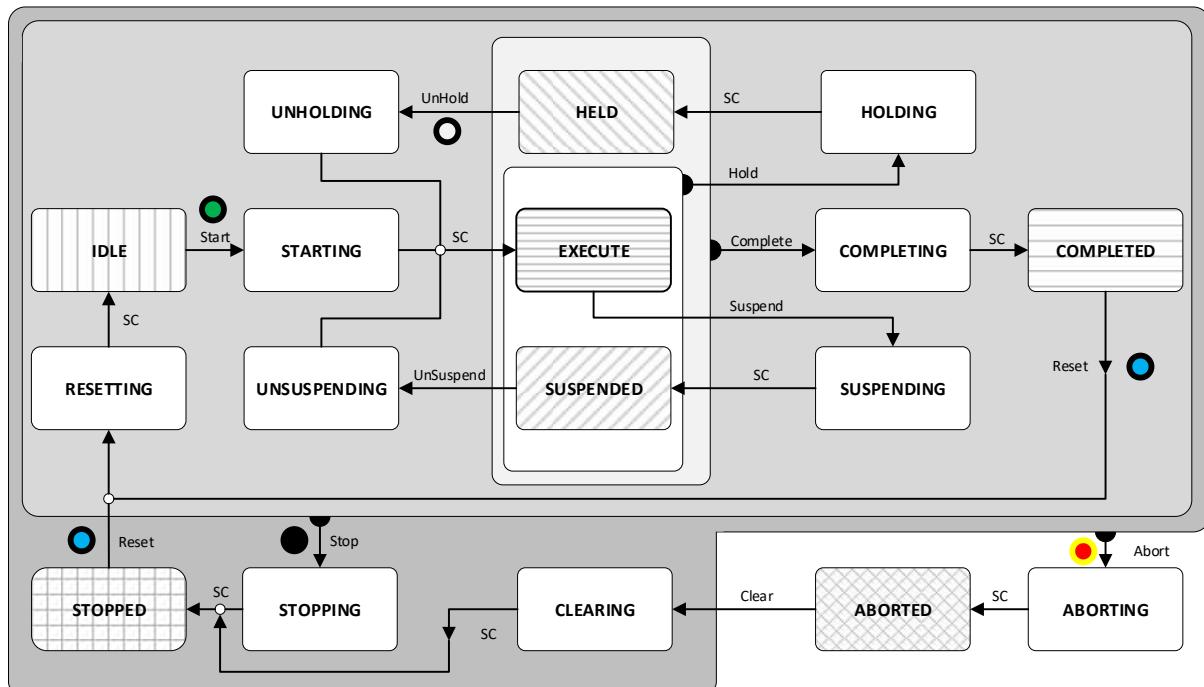


图 11：从 HMI 操作¹¹机器的按钮。

关于按钮的详细信息，请参见 PackML Guideline: Part 6: PackML User Interface - HMI。

¹⁰机器上按钮控制设备的颜色可根据标准 IEC 60204-1: 1997, 'Safety of Machinery' 定义：

- START 按钮的颜色可以是白色、灰色或黑色，但首选白色。也允许使用绿色。但严禁使用红色。
- 紧急停机按钮应使用黄色/红色。
- STOP 按钮的颜色可以是黑色、灰色或白色，但首选黑色。也允许使用红色，但建议不要在紧急操作设备附近使用红色。不应该使用绿色。
- 交替起着 START 和 STOP 按钮作用的按钮致动器首选白色、灰色或黑色。不应使用红色、黄色或绿色。
- 在按下时触发操作、在松开后停止操作的按钮首选白色、灰色或黑色（例如按住运行按钮）。不应使用红色、黄色或绿色。
- Reset 按钮应使用蓝色、白色、灰色或黑色。如果它们兼具 STOP 按钮的作用，可使用白色、灰色或黑色，其中首选黑色。不应该使用绿色。

¹¹需要操作员干预才能从 Aborted 状态转换为 Clearing 状态。需要一个 Clear 命令。

6.5 从内部和外部角度看待 PACKML STATE MODEL

图 12 显示了 Unit/Machine 生产时的两种不同停止情况，即 HELD 情况和 SUSPEND 情况。

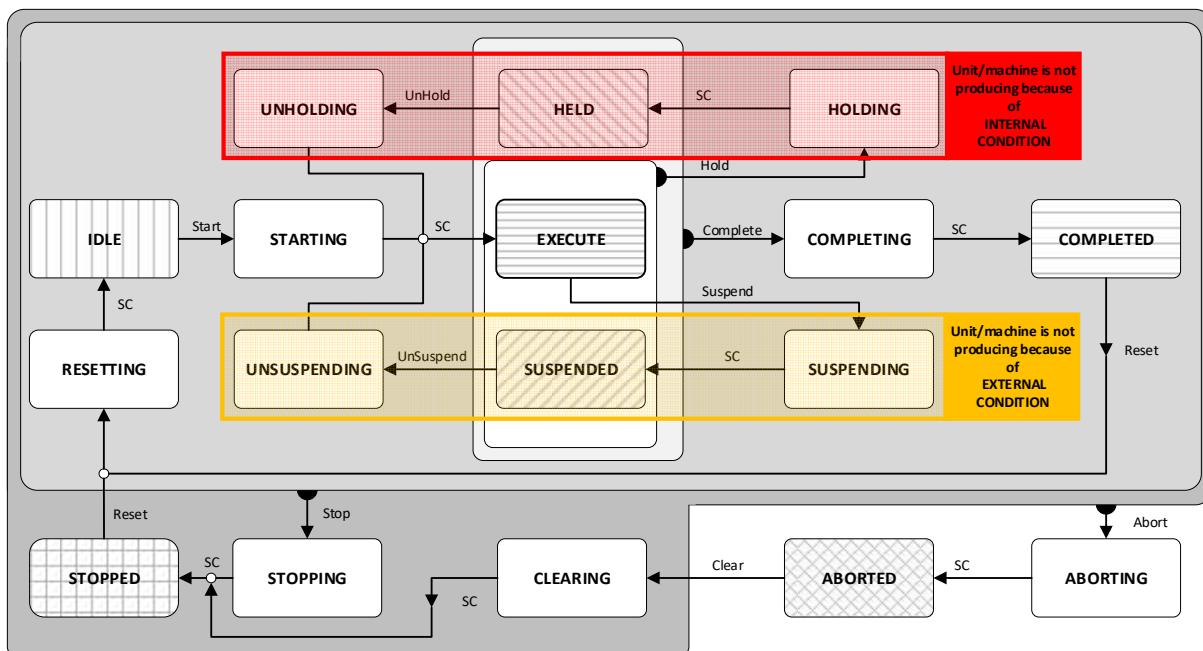


图 12：生产模式下 HELD 和 SUSPEND 的不同。

HOLDING
HELD
UNHOLDING

内部 Unit/Machine 条件不允许 Unit/Machine 继续生产时应使用这些状态。Unit/Machine 因内部条件退出 EXECUTE 或 SUSPENDED 状态时就会发生这种情况。注意内部机器条件是指 Unit/Machine 内部的条件，而不是来自生产线上其他机器的条件。

HELD 状态通常是指机器在日常运行过程中会出现的状况，而且它们要求操作员开展少量检修工作，随后机器就能恢复生产。HOLDING 状态可自动触发，也可由操作员手动触发，并可轻松恢复。

举例来说，Unit/Machine 要求操作员定期在点胶机中加注胶水或补充纸箱就属于此类条件，而且因为机器设计原因，这些操作无法在 Unit/Machine 运行期间执行。由于此类任务是生产操作的一部分，不宜通过终止或停机程序来执行，而且因为这些功能是 Unit/Machine 的有机组成部分，因此它们并不被视为外部条件。在 HOLDING 状态下，机器通常会受控停机，然后在状态完成后切换到 HELD。

提示：为了能在 HELD 状态后妥善地恢复生产，执行 HOLDING 程序时，必须将收到 HOLD 命令时所有相关工序设置点和状态信息保存到 Unit/Machine 控制器中。

提示：导致从 EXECUTE 转换为 HOLDING 的事件可能需要在 UNHOLDING 状态下重置。

SUSPENDING
SUSPENDED
UNSUSPENDING

外部 Unit/Machine 条件不允许 Unit/Machine 继续生产时应使用这些状态。外部工序条件是指 Unit/Machine 之外（但通常位于同一个集成生产线中）的条件。

Unit/Machine 因生产线上游或下游条件退出 EXECUTE 状态时就会发生这种情况。此类条件通常是由于堵塞或缺料事件导致的。此条件可由本地机器传感器检测，或由监控系统外部命令触发。在 SUSPENDING 状态下，Unit/Machine 通常会受控停机，然后在状态完成后切换到 SUSPENDED。

提示：为了能在 SUSPENDED 状态后妥善地恢复生产，执行 SUSPENDING 程序时，必须将收到 Suspend 命令时所有相关工序设置点保存到 Unit / Machine 控制器中。

提示：导致从 EXECUTE 转换为 SUSPENDING 的事件可能需要在 UNSUSPENDING 状态下重置。

6.6 从 UNIT/MACHINE 参数角度看待 PACKML STATE MODEL

- 生产订单

图 13 显示了各个状态和 PackTag 类型对应的信息交换。图中顶部的黑框表示要交换的信息。蓝色箭头表示与 PackML Status Tag 有关的数据，而红色箭头表示与 PackML Command Tag 有关的数据。绿色箭头表示与 PackML Admin Tag 有关的信息。

与 Unit/Machine 的数据通信由以下类型的数据构成，而每种数据都由相应的 PackTag 类型标识：

- Command Tag - 用于控制 Unit/Machine。
- Status Tag - 用于从 Unit/Machine 获取状态信息。
- Admin Tag - 用于从 Unit/Machine 获取性能信息。

表 2：数据类型和 PackTag 类型

数据类型	PackTag 类型
Mode 和 State	Status Tag
警告和报警	Admin Tag
性能数据 - OEE 数据、加工品、缺陷品、消耗品的计数	Admin Tag
命令 - Reset、Start、Stop、Complete 等	Command Tag
参数（机器参数）	Command Tag Status Tag
配方（控制配方参数）	Command Tag Status Tag

各单元的控制逻辑和同步通过各种 Command Tag 来管理。命令用于请求更改单元的状态。命令可由内部 Unit/Machine 条件、Unit/Machine HMI（按钮）触发，也可从外部系统接收。

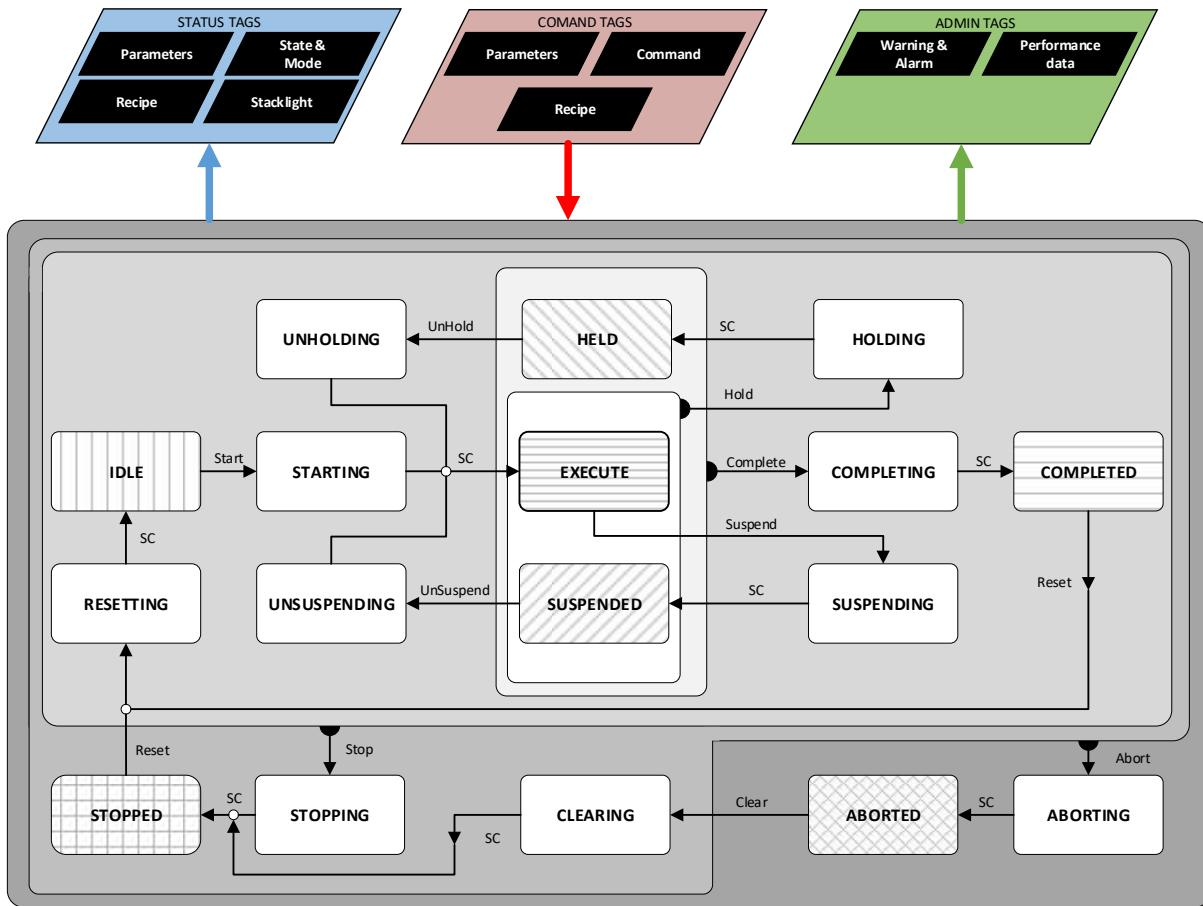


图 13：从 Unit/Machine 参数角度看待 PackML State Model

提示: 在 PackML Interface State Manager 的设计中，建议指定 PackML Interface State Manager 与供应商专用 Machine State Manager 之间交换哪些信息以及何时交换信息。

为了简化通信和映射，建议指定两个 State Manager 之间何时可以发送和接收信息以及在不同状态下可以交换哪些信息。例如，只有在 IDLE 或 STOPPED 状态下才能在 PackML Interface State Manager 与供应商专用 Machine State Manager 之间交换 Command Tag 参数。通常来说，操作员也可以更改相关的 Unit / Machine 参数。

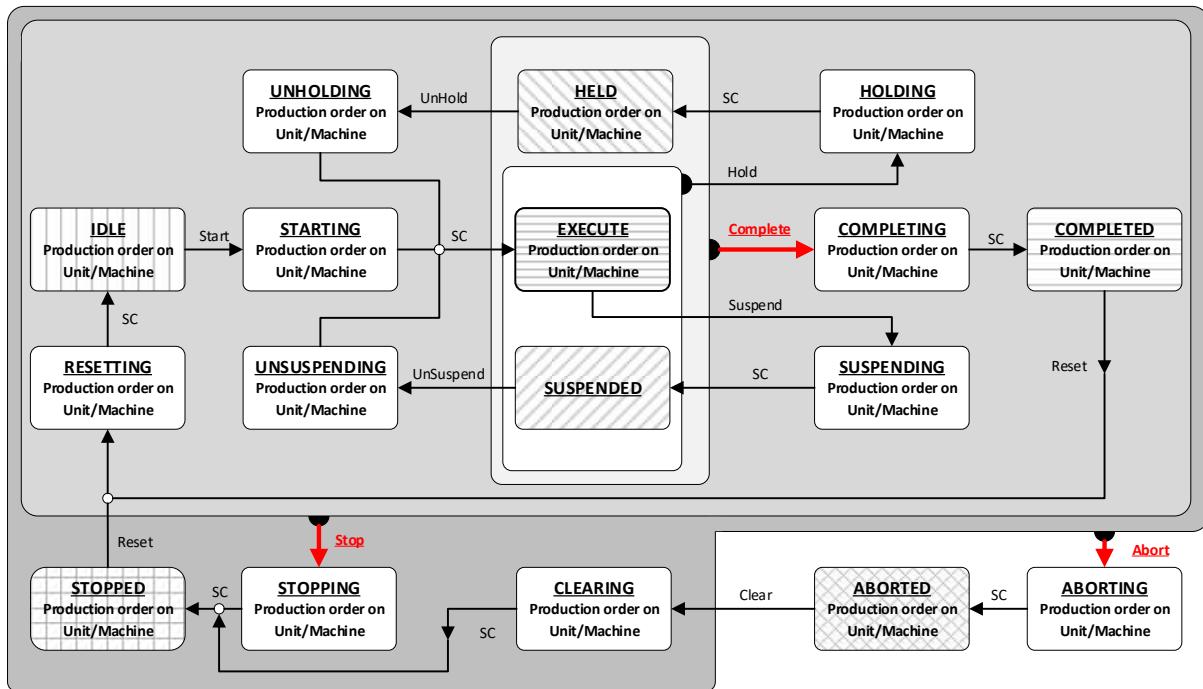


图 14：从生产订单角度看待 PackML State Model¹²

PackML 基本状态模型源于批次生产行业（参见 Ref. No. S88-1）而且该状态模型反映了执行生产订单（批次）时的变化。

终止生产订单有三种方式：

- 通过 Stop 命令：意味着当前生产的产品可以使用。生产订单停止。
- 通过 Complete 命令：意味着当前生产订单已按照定义的标准（例如，生产的产品数量）结束。生产订单完成。
- 通过 Abort 命令：意味着当前生产的产品可能已损坏。生产订单终止。

¹²使用生产订单，因为它代表参考 ISA-88.00.01 中定义的 Unit/Machine 内的产品处理。

6.7 PACKML STATE MODEL 和堆栈灯

Status Tag Status.Stacklight[#] 用于指示与单个 PackML State 相关的堆栈灯状态。

Status.Stacklight[#] Tag 是一个 DINT，单个 Bit 位置与特定的堆栈灯颜色和功能（即常亮或闪烁）相关。

Bit 编号	堆栈灯功能
0	红色亮灯
1	红色闪烁
2	黄色亮灯
3	黄色闪烁
4	蓝色亮灯
5	蓝色闪烁
6	绿色亮灯
7	绿色闪烁
8	橙色亮灯
9	橙色闪烁
10..31	用户-定义

在 ISA TR88.00.02-2022 中，举例说明了如何使堆栈灯对应于某些机器条件和 PackML State。该示例符合标准 IEC 60073，也符合 OMAC 指南的 Part 6: PackML User Interface - HMI。

	Aborting	Clearing	Aborted	Complete	Stopped	Resetting	Idle	Starting	Execute	Completing	Holding	Held (No Production)	Unholding	Suspending	Suspended	Unsuspending	Stacklight	Status.Stacklight[#]
Abnormal Stop	F	F	F														Red Lamp Flashing	1
Controlled Stop		S	S	S	S												Red Lamp Solid	0
Starved Upstream													F	F			Amber Lamp Flashing	3
Blocked Downstream												S	S				Amber Lamp Solid	2
Low Material	F	F	F	F	F	F	F	F	F		F	F	F	F		Blue Lamp Flashing	5	
Material Exhausted										S	S						Blue Lamp Solid	4
Ready to Start						F											Green Lamp Flashing	7
Running							S	S	S			S			S	Green Lamp Solid	6	
Starting / Restarting								F			F				F	Horn Flashing	9	
Not used																Horn Solid	8	
Not used																User-Defined	10..31	

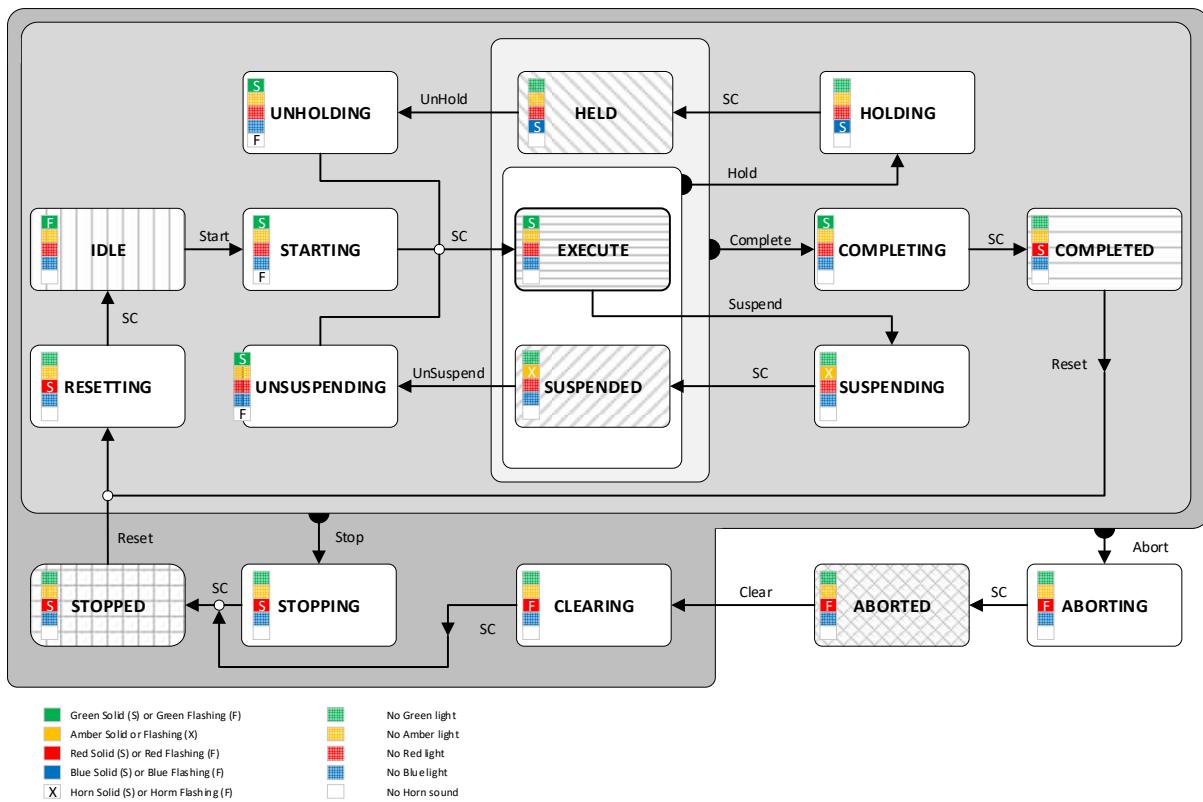


图 15：PackML State Model 和堆栈灯示例。

7. PACKML EVENT STATE MANAGER

我们可以制作一个对照表，以方便机器供应商配置哪些类型的 Unit/Machine 事件会导致 PackML Interface State 变化，和哪些 PackML Interface State 变化会导致 Unit/Machine 事件。据此我们就可以确定某个 Unit/Machine 何时应该切换到 PackML Interface State Model 中的特定状态。

例如，当 Unit/Machine 进行内部状态更改并映射到 PackML Interface State: HELD、STOPPED、SUSPENDED 和 ABORTED 时，最终用户可能对映射有不同的解释和要求。第 7 节以下部分中的的表格表示 PackML Interface State Manager 和供应商专用 Machine State Manager 之间的映射关系的示例。在各表中，表格左侧（起始行为蓝色）表示 PackML Interface State，表格右侧（起始行为灰色）表示供应商专用 Machine State，颜色匹配与图 16 相对应。表格下方的一组插图显示了 Acting PackML Interface State 与示例供应商专用 Machine State 之间所有可能的映射关系。供应商专用 Machine State 使用标准 PackML State Model 说明，但需要注意的是，机器供应商的状态模型可能十分不同。

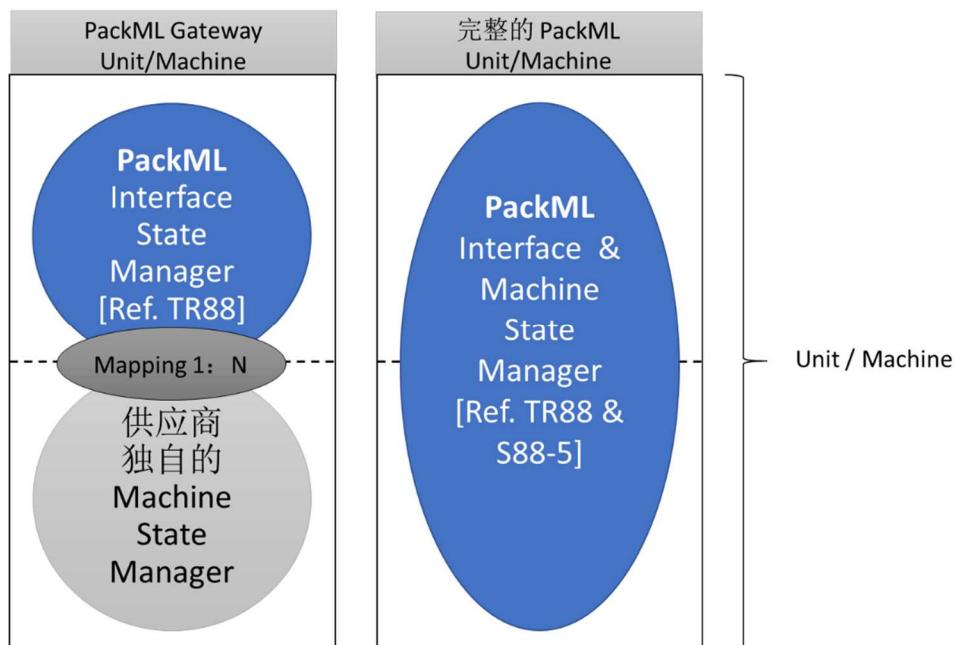


图 16: PackML Interface State Manager 和 PackML 不同 Machine State Manager (供应商专用或 PackML) 之间的映射关系

提示: 机器的买方和卖方必须就机器何时必须处于哪个 PackML interfaces state 达成一致。建议使用一个对照表，其中将 PackML Interface State 作为该表格的一个维度，将当前机器专用 Machine State 作为该表格的另一个维度。或作为另一个维度使用的 Unit/Machine tag 需要映射到当前 PackML State。同时，列出不同 PackML Interface state 下所有可能的情况。有关详细信息，请参见第 7.11 节的示例对照表

7.1 RESETTING

表 3：Resetting 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
RESETTING	<p>PackML interface RESETTING 状态将启动复位程序，这通常会导致 Unit/Machine 清除数据并将 Unit/Machine 置于 IDLE 状态，在此状态下 Unit/Machine 组件处于通电状态下并等待 Start 命令。将单元复位后，机器参数将被清除，而且 Unit/Machine 将等待新的机器参数。例如，操作员已更换工具，而通过激活 Resetting 程序，表明机器做好了生产准备。</p> <p>在 RESETTING 状态下要执行的动作须由机器供应商和最终用户指定。</p>	Resetting	<p>此状态是在 STOPPED 或 COMPLETE 状态下执行 RESET 命令的结果。这时 Fault 和 Stop 原因都会被复位。RESETTING 通常会激活安全设备，并将机器置于 IDLE 状态下并等待 START 命令。</p> <p>RESETTING 通常会在 Unit/Machine 重置后复位激活安全设备引起的报警。</p> <p>在此状态下不应出现危险动作。</p> <p>此动作状态由复位请求  触发。</p> <p>在此状态下，机器可能会在停机后移动，例如“re-homing”¹³。</p>

¹³ 对供应商专用 Machine State Manager 来说，在合适的时间清除数据至关重要，以确保 Unit/Machine 不会在装有错误生产订单的工具和组件的情况下物理启动。

PACKML INTERFACE (RESETTING)

生产订单/批次被 Reset，这时可启动新的生产订单/批次。

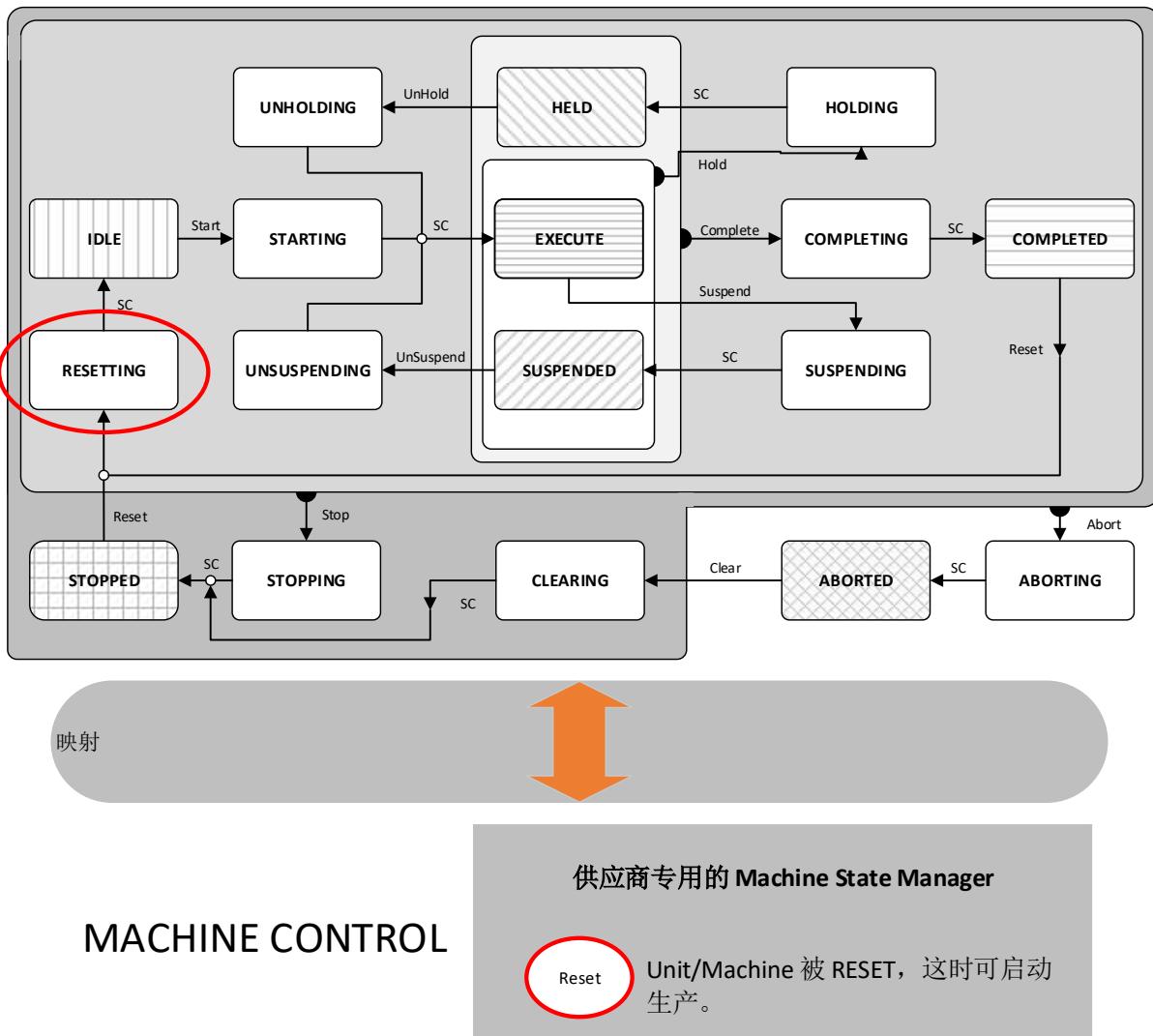


图 17: PackML Interface State Model 和 Machine control State model 之间的 Reset 映射关系¹⁴。

¹⁴ Unit/Machine 上的 RESET 并不是在监管控制层级和生产线中的其他单元上“结束生产订单”。下游单元可能仍在执行监管控制层级的生产订单。

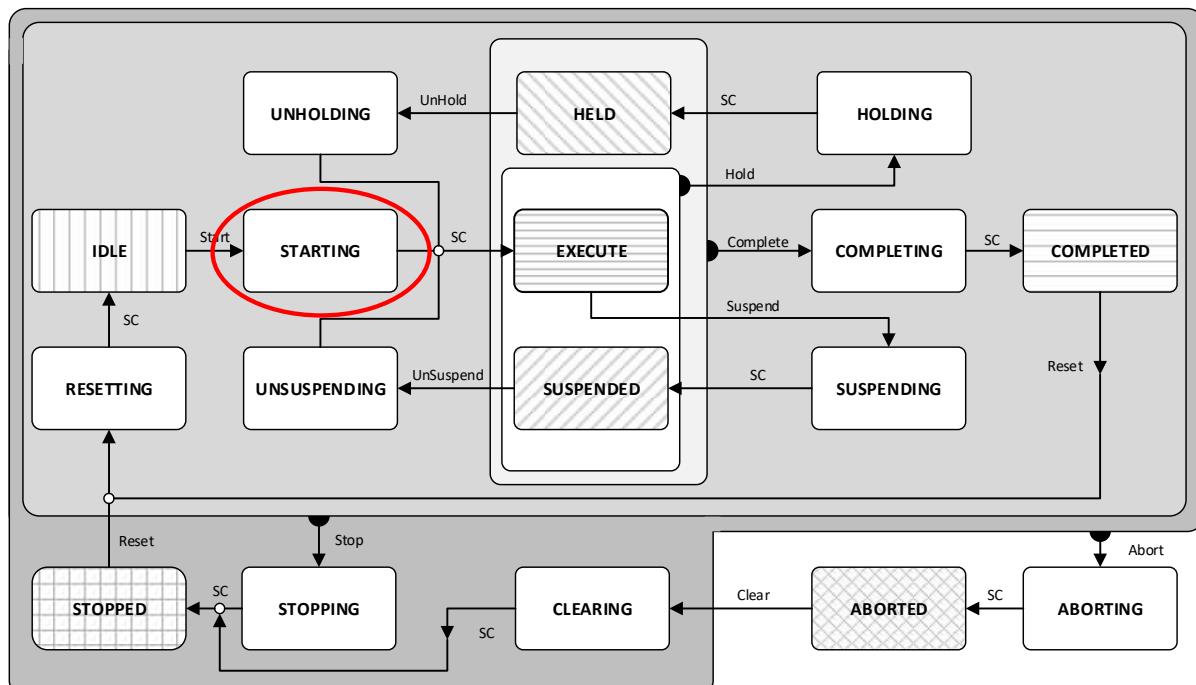
7.2 STARTING

表 4: Starting 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
STARTING	<p>此状态提供启动 Unit/Machine 所需的步骤，而且由 Start 命令触发。该启动逻辑将检查机器参数，如这些参数都经过批准，则将 Unit/Machine 切换到 EXECUTE 状态。该启动逻辑将逐渐提升速度、压力等，从而为生产做好准备。但是，如果参数无效，该启动逻辑会将 Unit/Machine 切换到 STOPPING 状态，随后 Unit/Machine 将止于 STOPPED 状态。</p> <p>如果机器无法完成 Starting 状态处理（比如因为内部错误），Unit/Machine 将进入 STOPPING 状态，并止于 STOPPED 状态。</p> <p>在 STARTING 状态下要执行的动作须由机器供应商和最终用户指定。</p>	Starting	<p>机器完成 START 所需的步骤。进入此状态是 Start 命令（本地或远程）的结果。在此命令后，机器将开始“execute”。发出 Starting 请求 ● 的结果是，机器将进入 EXECUTE 状态。</p>

PACKML INTERFACE (STARTING)

生产订单/批次正在启动。



映射



MACHINE CONTROL

供应商专用的 Machine State Manager

Starting

Unit/Machine 启动，并为执行生产
订单/批次做好了准备。

图 18: PackML Interface State Model 和 Machine control State model 之间的 Starting 映射关系

7.3 EXECUTE

表 5: Execute 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
EXECUTE	Unit/Machine 正在运行，而且所选 MODE 和/或所选配方定义的所有条件都满足。	Execute	<p>机器处理物料期间，它即处于 EXECUTE 状态。不同的机器模式会导致特定类型的 EXECUTE 活动。</p> <p>例如，如果机器处于“生产”模式下，EXECUTE 活动为生产产品。机器此时会按照设置的速度生产产品。</p> <p>提示：最好将设置的速度存储为产品“配方”中的控制配方参数，以免由于不必要的阻塞和缺料造成的速度损失。</p>

PACKML INTERFACE (EXECUTE)

生产订单/批次正在启动。

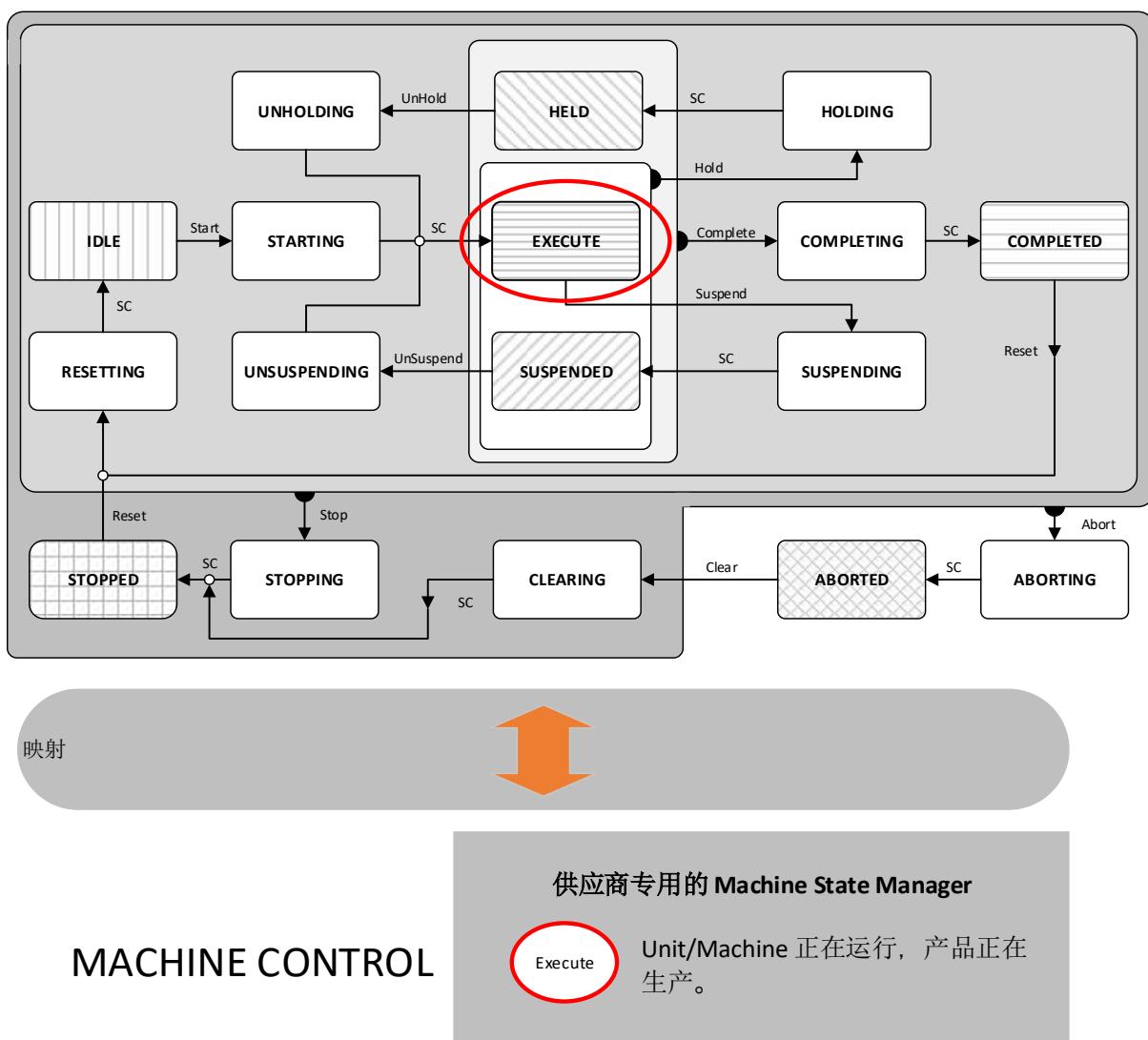


图 19: PackML Interface State Model 和 Machine control State model 之间的 Execute 映射关系

7.4 HOLDING

以下是可以将 PackML Interface State Model 置于 HOLDING 状态的三种机器控制场景。

Unit/Machine 可以因为以下原因之一处于 HOLDING 状态：

- 1) Unit/Machine 上的挂起条件导致的小问题
- 2) Unit/Machine 停机，但不妨碍继续当前生产订单
- 3) 或者要求 Unit/Machine 安全停机的大问题（ABORTING），但不妨碍继续当前生产订单

通常来说，Unit/Machine 只支持上述三种场景中的一种。

表 6: Holding 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
HOLDING	HOLDING 逻辑会令 Unit/Machine 停机，或者会将特定单元置于代表 HELD 的状态。这是 Unit/Machine 上出现错误 ¹⁵ 时或操作员从 Unit/Machine HMI 启动 Hold 命令时执行的内部控制逻辑。 Holding 控制逻辑会将 Unit/Machine 从 HOLDING 状态切换为 HELD 状态。	Holding ¹⁶	<p>此状态应在内部机器条件（Unit/Machine 内部的条件，而不是来自生产线上其他机器的条件）不允许机器继续生产时使用，也就是说机器因内部条件而退出 EXECUTE 或 SUSPENDED，但不妨碍继续当前的生产订单。</p> <p>这通常是指机器在日常运行过程中会出现的状况，而且它们要求操作员开展少量检修工作，随后机器就能恢复生产。此状态可自动触发，也可由操作员手动触发，并可轻松恢复。举例来说，机器要求操作员定期在点胶机中加注胶水或补充纸箱就属于此类条件，而且因为机器设计原因，这些操作无法在机器运行期间执行。由于此类任务属于正常生产操作，不宜通过终止或停机程序来执行，而且因为这些功能是机器的有机组成部分，因此它们并不被视为外部条件。在 HOLDING 状态下，机器通常会受控停机，然后在状态完成后切换到 HELD。为了能在 HELD 状态后妥善地恢复生产，执行 HOLDING 程序时，必须将收到 HOLD 命令时所有相关工序设置点和流程恢复状态信息保存到 Unit/Machine 控制器中。</p>
		Stopping ¹⁶	<p>完成 STOPPING 状态后，机器通电却无动作，在纠正停止条件后可以继续当前生产订单。此时与其他系统的所有通信（如有）仍正常进行。</p> <p>发出受控停机请求的结果是，机器将进入 STOPPED 状态。</p>
		Aborting ¹⁶	<p>响应 ABORT 命令时或发生机器故障时会随时进入 ABORTING 状态。ABORTING 逻辑会令机器迅速地安全停机，在纠正中止条件后可以继续当前生产订单。</p>

¹⁵ 并非所有机器错误都会导致机器切换到 Holding 状态。例如，如果某个错误并不影响 Unit/Machine 上的生产任务，则不应将 Unit/Machine 置于 PackML interface state HOLDING 状态。OEM 可以编制一张表格，说明哪些 Unit/Machine 错误和事件应将 Unit/Machine 置于特定的 PackML Interface State。错误和事件应映射到报警和警告信息。

¹⁶ PackML Interface State 和供应商专用 Machine State Manager 之间的映射关系的示例，参阅第 7.11 节。

PACKML INTERFACE (HOLDING 源自内部 hold)

将 Unit/Machine 切换到 HOLDING state 的原因有以下几种:

- 1)Unit/Machine 上的挂起条件导致的小问题

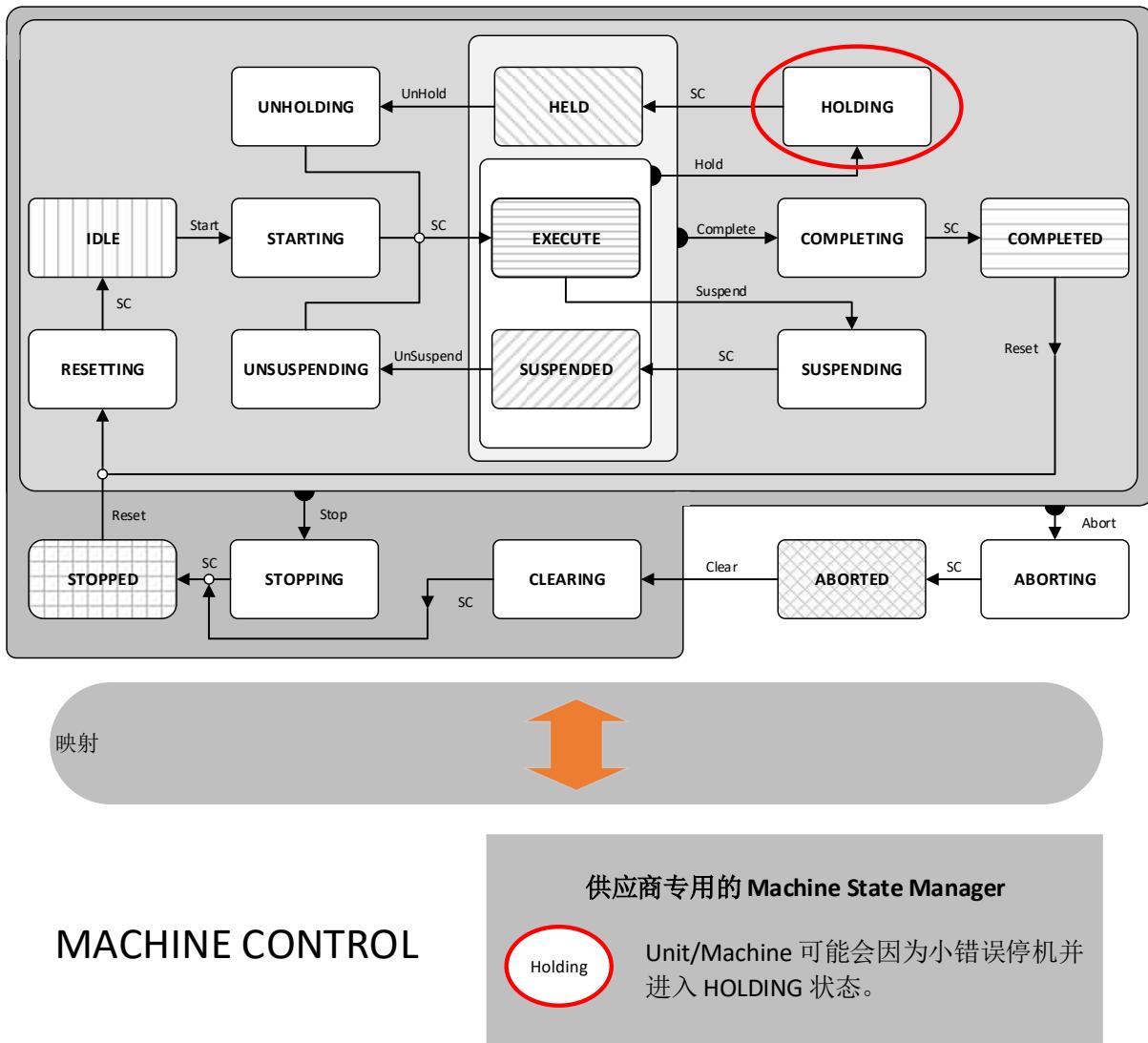


图 20: PackML Interface State Model 和 Machine control State model 之间的 Hold 映射关系

PACKML INTERFACE (HOLDING 源自内部 stop)

将 Unit/Machine 切换到 HOLDING state 的原因有以下几种:

- 2) Unit/Machine 停机

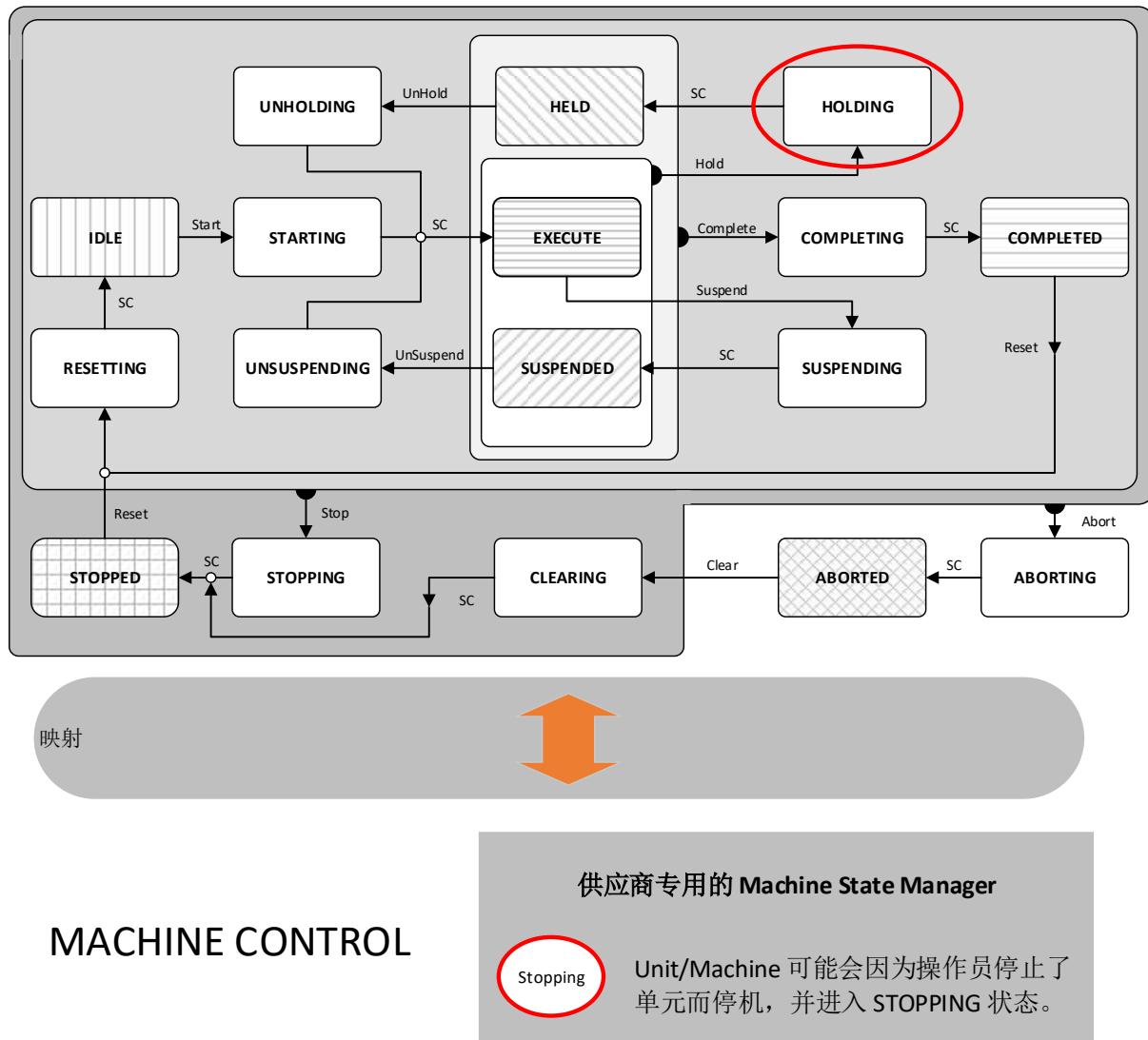


图 21: PackML Interface State Model 和 Machine control State model 之间的 Stopping 映射关系

PACKML INTERFACE (HOLDING 源自内部 abort)

将 Unit/Machine 切换到 HOLDING state 的原因有以下几种：

- 3) 或者要求安全停机的大问题

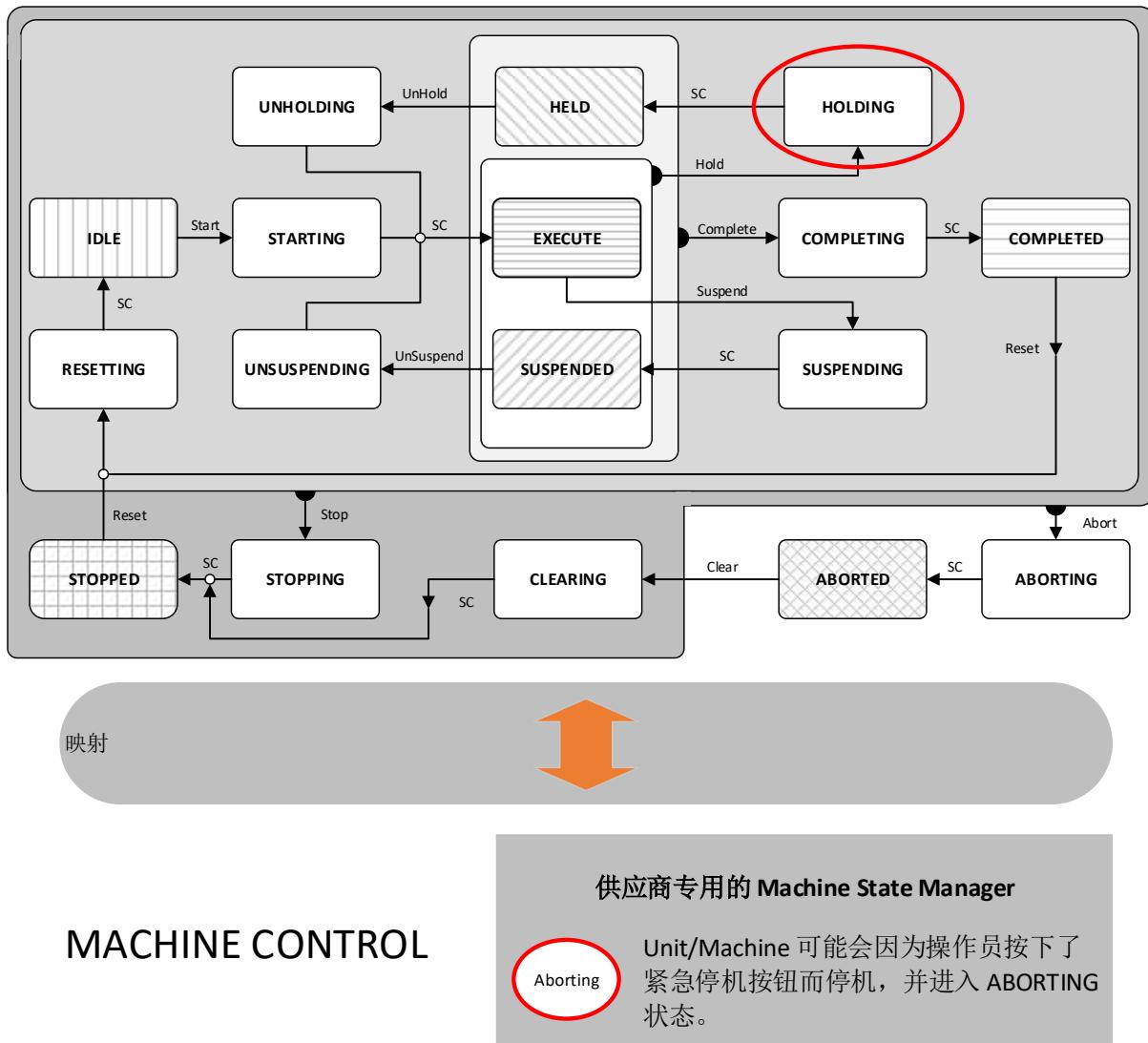


图 22: PackML Interface State Model 和 Machine control State model 之间的 Aborting 映射关系

7.5 UNHOLDING

UNHOLDING 场景与 Unit/Machine 进入 HOLDING 状态的方式有关。HOLDING 状态具有三种不同的场景，因此 UNHOLDING 处理也有三种不同的场景。UNHOLDING 处理会将 Unit/Machine 切换到 EXECUTE 状态。

- 1) 执行 UNHOLDING 可以很简单，只需从机器控制系统重新启动 (Unhold) Unit/Machine 即可。
- 2) 机器控制系统原本已停止，因此需要在同一生产订单上重新启动。
- 3) 机器控制系统原本已紧急停止，因此需要清除状态并在同一生产订单上重新启动。

表 7: UNHOLDING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
UNHOLDING	UNHOLDING 状态是对操作员执行的恢复生产命令的响应。UNHOLDING 状态下的控制逻辑会让 Unit/Machine 做好重新进入正常 EXECUTE 状态的准备。该 UNHOLDING 逻辑将逐渐提升速度、压力等，从而为生产做好准备。	Unholding	请参见 HOLDING 了解该状态的使用场景。机器通常会在内部条件（例如物料水平）恢复到可接受的水平时自动进入 UNHOLDING 状态。如果需要操作员开展少量检修工作来补充物料或执行调整，则可由操作员启动 UNHOLD 命令。
		Resetting Idle Starting	操作员需要在 Machine Control panel (HMI) 上开展操作来恢复至 EXECUTE 状态。操作员此前令 Unit/Machine 受控停机，那么在此状况后需要在同一生产订单上重新启动。
		Clearing Stopped Resetting Idle Starting	操作员此前激活了安全回路，令 Unit/Machine 急停。操作员需要清除安全回路，并在 Machine Control panel (HMI) 上开展操作来恢复至 EXECUTE 状态。操作员此前紧急停止了 Unit/Machine，现在需要清除紧急停止回路并在同一生产订单上重新启动。

PACKML INTERFACE (UNHOLDING 源自 unholding)

由于修复了 Unit/Machine 上的问题，生产订单/批次执行操作现在 UNHOLDING。

这通常是指解决了某个小问题。机器控制器并未停止。

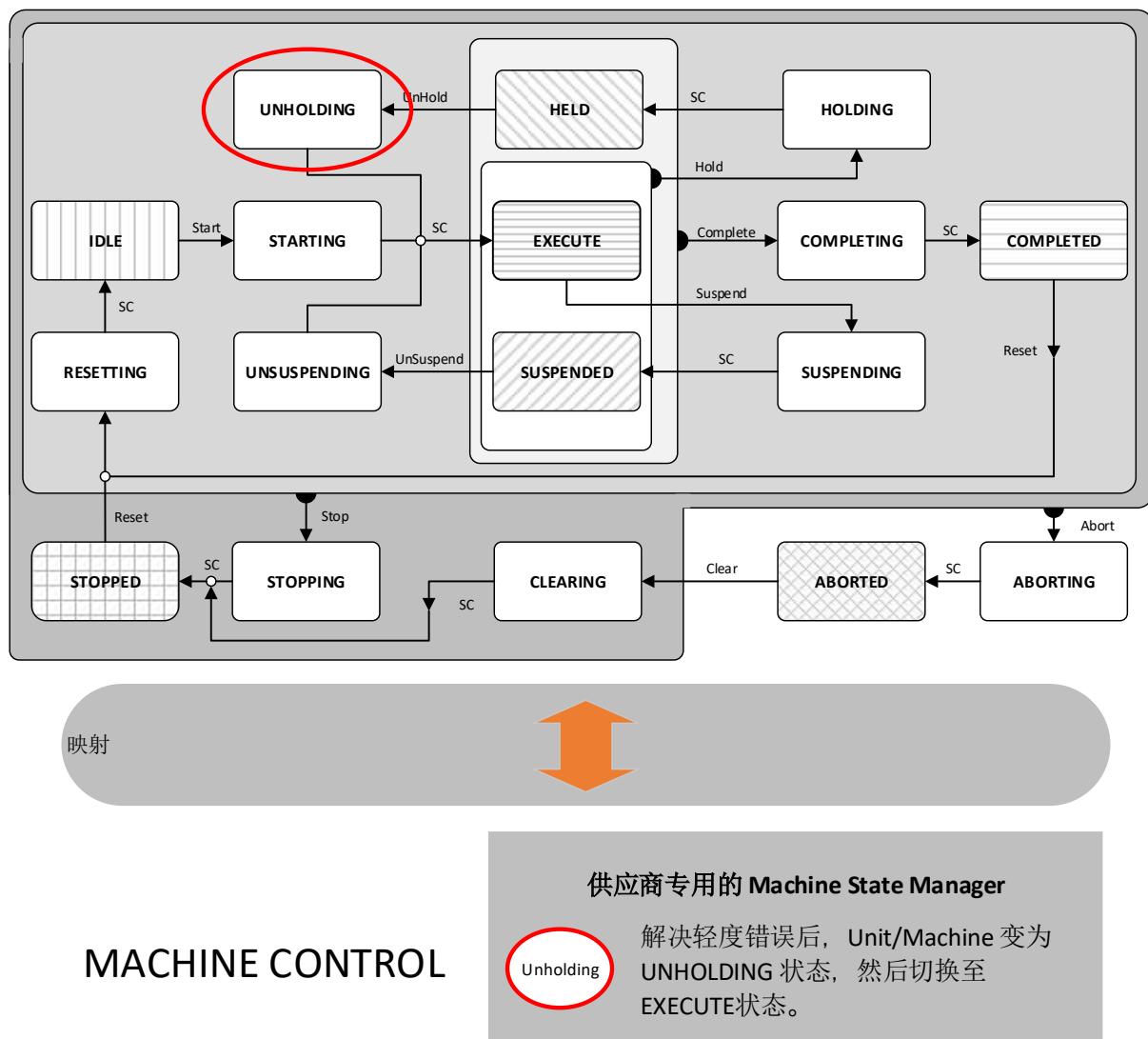
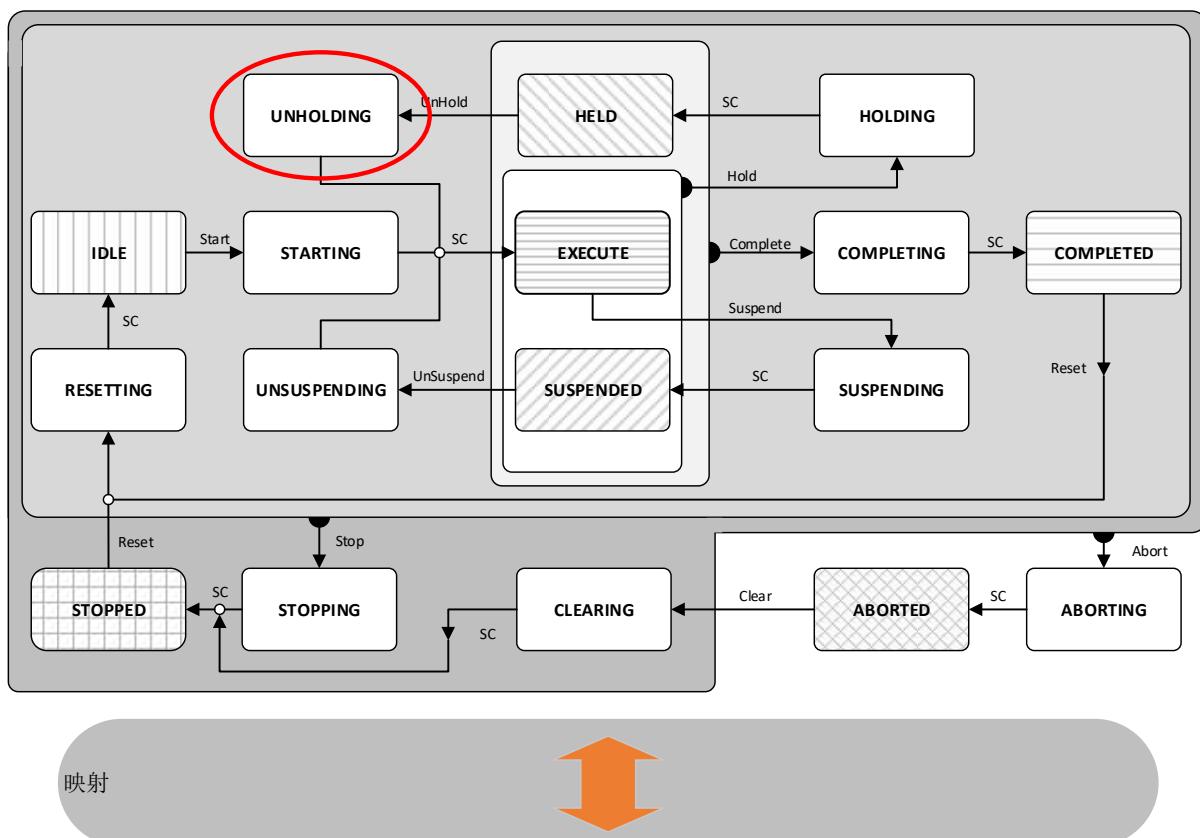


图 23：PackML Interface State Model 和 Machine control State model 之间的 Unhold 映射关系

PACKML INTERFACE (UNHOLDING 源自 resetting、idle 和 starting)

由于修复了 Unit/Machine 上的问题，生产订单/批次执行操作现在 UNHOLDING。

机器控制器已停止，现在需要重新启动。



MACHINE CONTROL

供应商专用的 Machine State Manager

Resetting
Idle
Starting

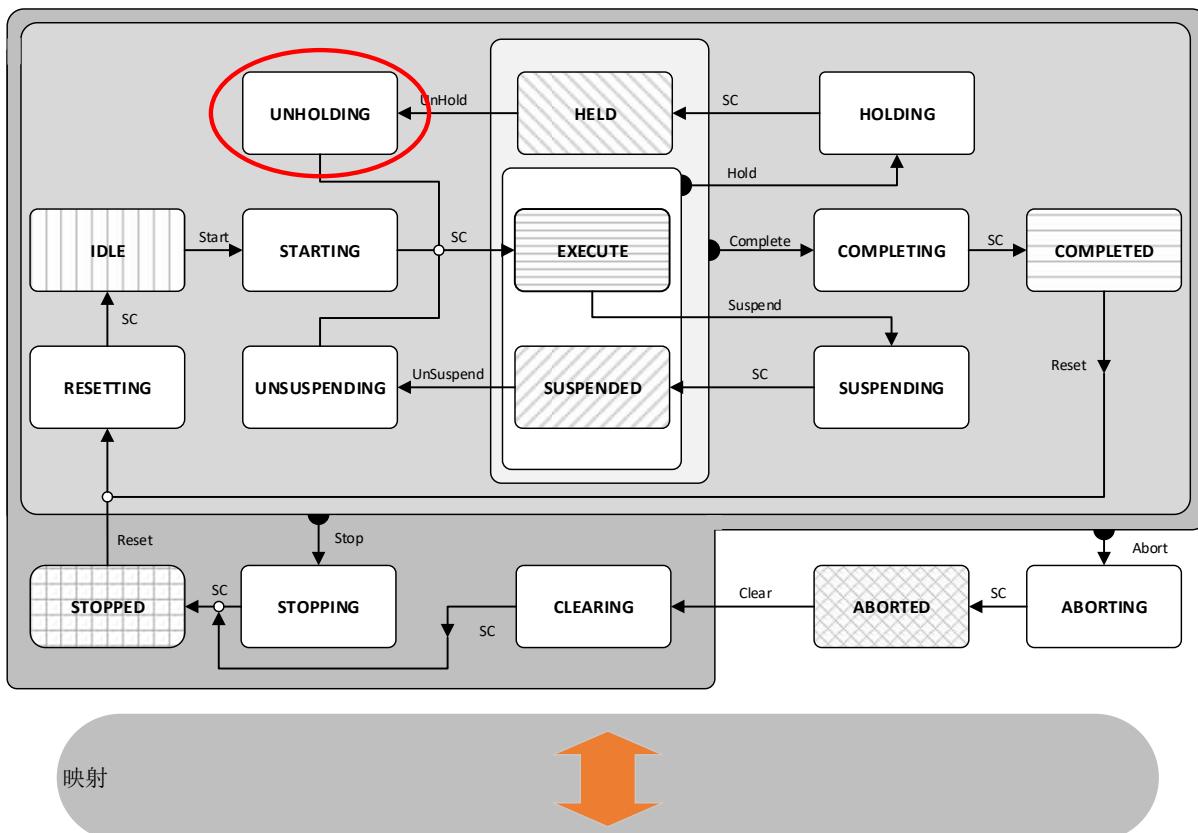
如果操作员在RESET Unit/Machine
后按下START, Unit/Machine
将启动。unit 切换至 EXECUTE 状态。

图 24: PackML Interface State Model 和 Machine control State model 之间的 Resetting 映射关系

PACKML INTERFACE (UNHOLDING 源自 clear、stopped、...)

由于修复了 Unit/Machine 上的问题，生产订单/批次执行操作现在 UNHOLDING。

机器控制器已紧急停止，现在需要清除并重新启动。



MACHINE CONTROL

供应商专用的 Machine State Manager

Clear
Stopped
Resetting
Idle
Starting

如果操作员在 RESET Unit/Machine 后按下 START, Unit/Machine 将启动。unit 切换至 EXECUTE 状态。

图 25: PackML Interface State Model 和 Machine control State model 之间的 Clear 映射关系

7.6 SUSPENDING

以下是可以将 PackML Interface State Model 置于 SUSPENDING 状态的两种机器控制场景。

Unit/Machine 可以因为以下原因之一处于 SUSPENDING 状态：

- 1) 上游或下游缺料或物料饱和（Blockage）。
- 2) 由于外部条件，操作员已停止机器控制。

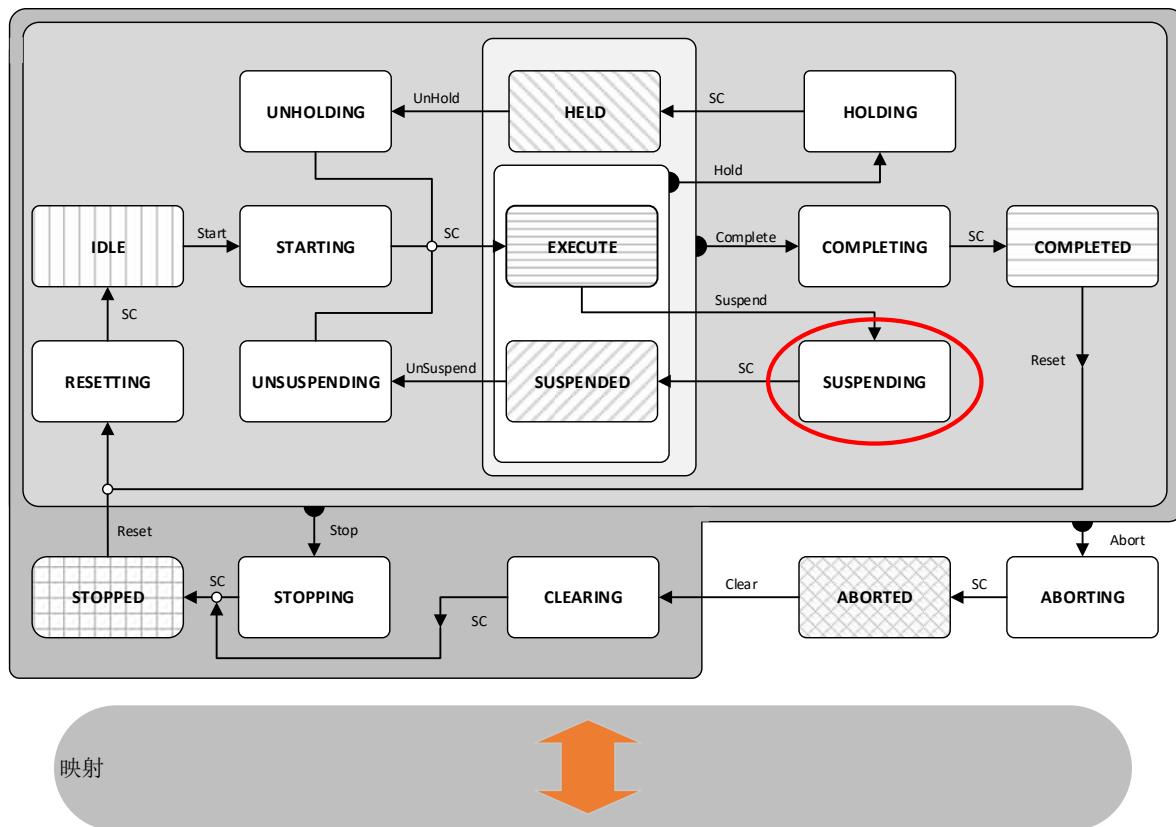
表 8: SUSPENDING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
SUSPENDING	<p>SUSPENDING 状态是进入 SUSPENDED 等待状态前必须经过的一个状态，它会让 Unit/Machine 为停止熨烫、停止切割、停止除袋等操作做好准备。</p> <p>此时 Unit/Machine 可能在相关的设置速度下运行，但没有产品被发送到下游的下一个 Unit/Machine。¹⁷</p> <p>机器供应商和最终用户必须指定机器进入 Suspending 和 SUSPENDED 状态时要发生的特殊状况。例如，Unit/Machine 应该装满内部产品缓冲区，还是应该立即停机。</p> <p>Unit/Machine 进入 Suspending 状态时要执行的操作需要定义。</p> <p>Suspending 是进料区缺料或出料区物料饱和，以致于 Unit/Machine 无法 Executing 继续稳定的生产。执行受控的 Suspending 程序期间，Unit/Machine 将切换至 SUSPENDED 状态。</p>	Suspending	<p>此状态应在外部工序条件（本 Unit/Machine 之外，但通常还在同一条集成生产线上）不允许机器继续生产时使用。通常机器是因为生产线上游或下游条件而退出 EXECUTE。例如，如果上游设备无法供应物料或下游设备无法接收物料，Unit/Machine 将退出 EXECUTE。</p> <p>此条件可由本地机器传感器检测，或由监控系统外部命令触发。</p> <p>在 SUSPENDING 状态下，机器通常会受控停机，然后在状态完成后切换到 SUSPENDED。为了能在 SUSPENDED 状态后妥善地恢复生产，执行 SUSPENDING 程序时，必须将收到 SUSPEND 命令时所有相关工序设置点和流程恢复状态信息保存到 Unit/Machine 控制器中。</p>
	例如操作员需要休息时，操作员可通过外部接口强制触发 SUSPENDING 状态。	Stopping	<p>完成 STOPPING 状态后，机器通电，但无动作。此时与其他系统的所有通信（如有）仍正常进行。</p> <p>发出受控停机请求的结果是，机器将进入 STOPPED 状态。</p>

¹⁷必须定义各 Unit/Machine 上发生的情况，因为这是特定于 Unit/Machine 的。某些 Unit/Machine 可能不会因打开的产品而停止。

PACKML INTERFACE (SUSPENDING 源自 suspending)

如果在当前 Unit/Machine 之外的单元中发生了使当前 Unit/Machine 缺料或饱和的问题，则生产订单/批次的执行将变为 SUSPENDING 中。



MACHINE CONTROL

供应商专用的 Machine State Manager

Suspending

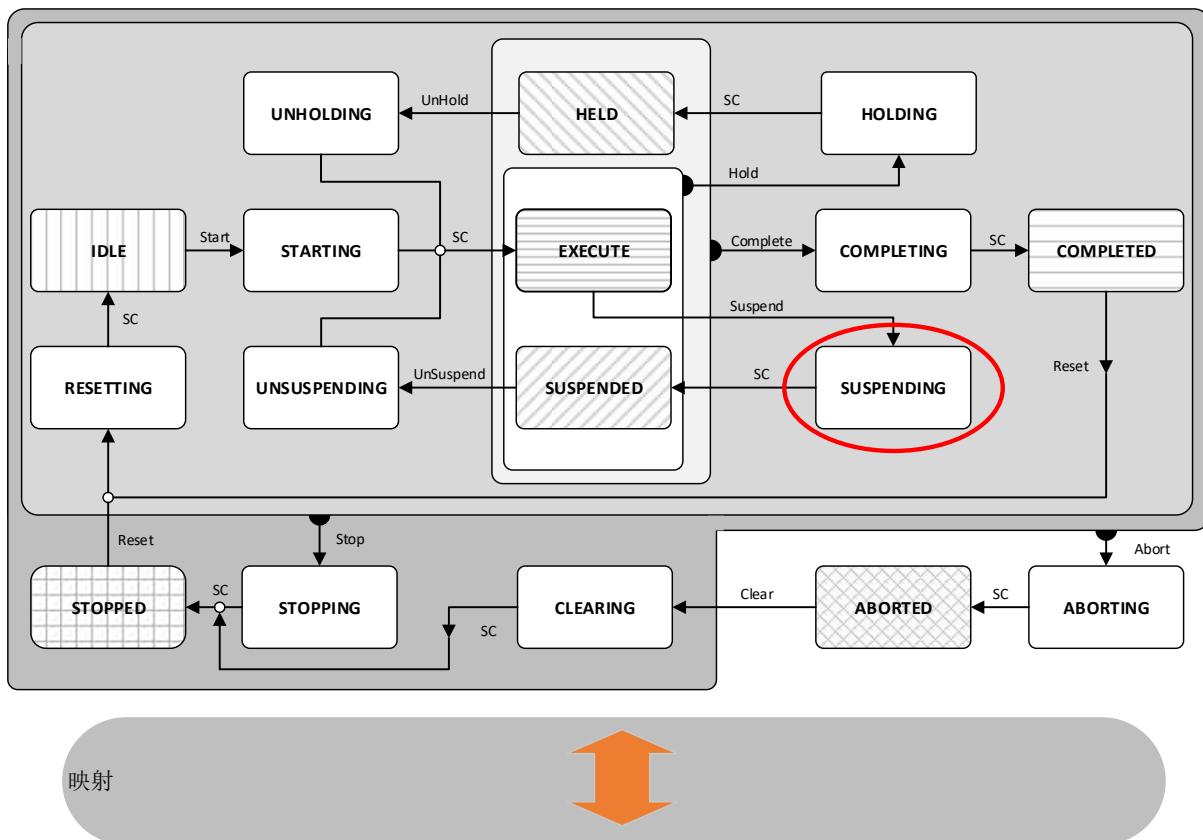
Unit/Machine 可因为外部问题
(例如缺料或产品堵塞) 而停机。

图 26: PackML Interface State Model 和 Machine control State model 之间的 Suspending 映射关系

PACKML INTERFACE (SUSPENDING 源自 stopping)

生产订单/批次的执行因为操作员执行了停机操作而 SUSPENDING。

例如，短暂的停歇。



MACHINE CONTROL

供应商专用的 Machine State Manager

Stopping

Unit/Machine 可因为操作员操作
而停机。

图 27: PackML Interface State Model 和 Machine control State model 之间的 Stopping 映射关系

7.7 UNSUSPENDING

UNSUSPENDING 场景与 Unit/Machine 进入 SUSPENDING 状态的方式有关。SUSPENDING 状态具有两种不同的场景，因此 UNSUSPENDING 处理也有两种不同的场景。

- 1) 机器控制级别自动变化，因为传感器指示上游或下游的缺料或物料饱和（Blockage）情况不再出现。
- 2) 操作员希望重新启动 Unit/Machine。

表 9: UNSUSPENDING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
UNSUSPENDING	<p>此状态是 Unit/Machine 在 SUSPENDED 状态下生成返回 EXECUTE 状态的请求或通过外部接口发出 Unsuspend 命令的结果。</p> <p>此状态在 EXECUTE 状态之前完成，并会让 Unit/Machine 做好进入 EXECUTE 状态的准备。</p> <p>此状态下的动作可能包括提升速度、打开真空以及重新吸合离合器。除非外部命令等于 Unsuspend，否则不允许 Unit/Machine 切换到 EXECUTE 状态。</p> <p>Unit/Machine 进入 UNSUSPENDING 状态时要执行的操作由机器供应商和最终用户定义。</p>	Unsuspending	<p>请参见 SUSPENDING 了解该状态的使用场景。此状态是工序条件恢复正常的结果。UNSUSPENDING 状态会启动所有必要的动作或必要的程序，将机器从 SUSPENDED 状态恢复至 EXECUTE 状态。为了能在 SUSPENDED 状态后妥善地恢复生产，执行 SUSPENDING 程序时，必须将收到 SUSPEND 命令时所有相关工序设置点和流程恢复状态信息保存到 Unit/Machine 控制器中。</p> <p>如果上游设备或下游设备可以重新接收产品，机器将退出 SUSPENDED 状态。例如，信号可能来自在生产线缓冲区变空时激活的外部生产线控制器或传感器，这时机器可重新恢复生产。</p>
		Resetting Idle Starting	操作员已在 Machine Control panel (HMI) 上开展操作来恢复至 EXECUTE 状态。操作员此前令 Unit/Machine 受控停机，已重新启动。

PACKML INTERFACE (UNSUSPENDING 源自 unsuspending)

由于修复了当前 Unit/Machine 以外机器上的问题，生产订单/批次执行操作现在 UNSUSPENDING。

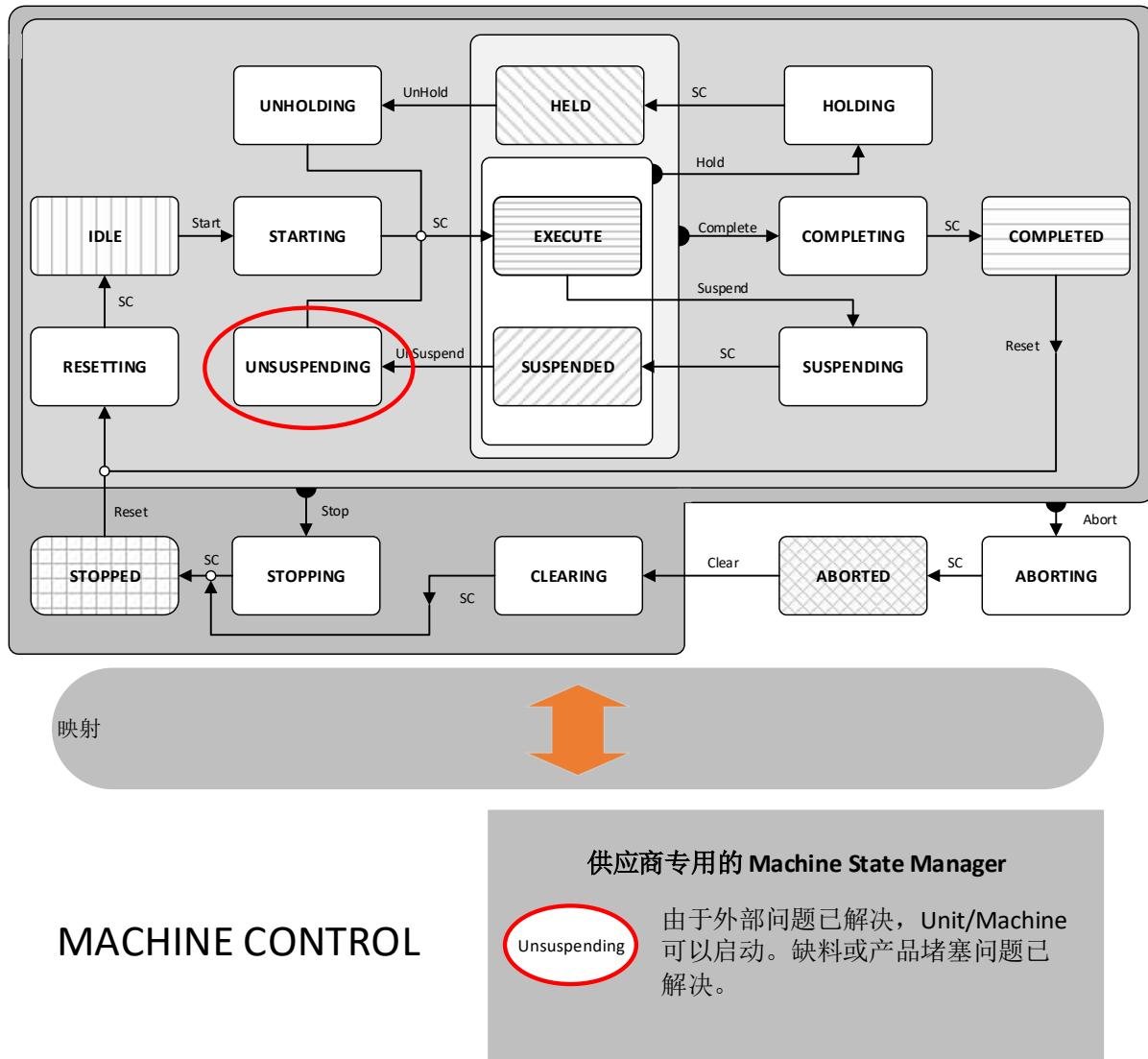


图 28: PackML Interface State Model 和 Machine control State model 之间的 Unsuspend 映射关系

PACKML INTERFACE (从 resetting 和 idle 切换为 UNSUSPENDING)

由于操作员希望重新恢复生产，生产订单/批次的执行变得 UNSUSPENDED。机器控制器已停止，现在需要重新启动。

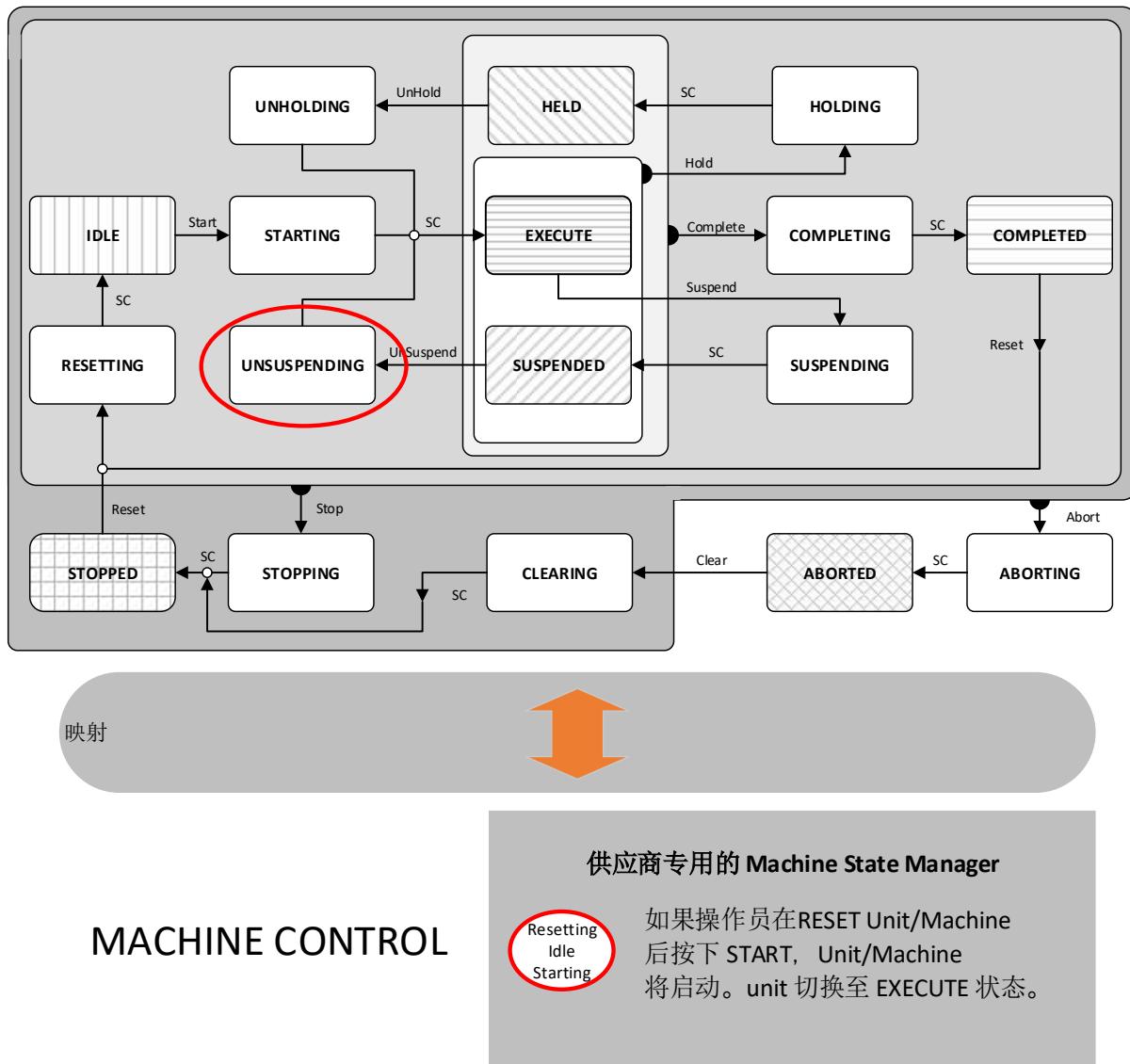


图 29: PackML Interface State Model 和 Machine control State model 之间的 Resetting 映射关系

7.8 COMPLETING

Completing 与 Unit/Machine 停止生产订单的方式有关。

表 10: COMPLETING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
COMPLETING ¹⁸	<p>此状态会提供必要的步骤来完成任务，通常是由于机器中的产品计数器达到了限制。Unit/Machine 会将机器参数提供给外部系统。机器参数可通过外部系统获取，从而了解执行期间在 Unit/Machine 上手动更改或更正的数据信息。</p> <p>机器供应商和最终用户需要指定在 Unit/Machine 进入 COMPLETING 和 COMPLETE 状态时要发生的特殊状况。例如，Unit/Machine 应该清空内部产品缓冲区，还是应该在机器中仍有产品时立即停机。</p> <p>在 COMPLETING 状态下要执行的动作须由机器供应商和最终用户指定。</p>	Completing	<p>当指定数量的产品已被处理时，此状态是对 EXECUTE 状态的自动响应。正常生产操作一直运行到结束，即进料区的物料处理将停止。</p> <p>例如，此状态也可由操作员在机器控制面板上按下 HMI 按钮来触发</p>

¹⁸ 当操作员希望强制“结束”或“完成”生产时。无法从外部系统进入 COMPLETING 状态。操作员必须向 Unit/Machine 发出 Stop 命令。这方便操作员通过外部系统在产品达到一定数量时灵活停止生产。

PACKML INTERFACE (COMPLETING 源自 completing)

由于生产了所需的产品数量，生产订单/批次执行 COMPLETING。

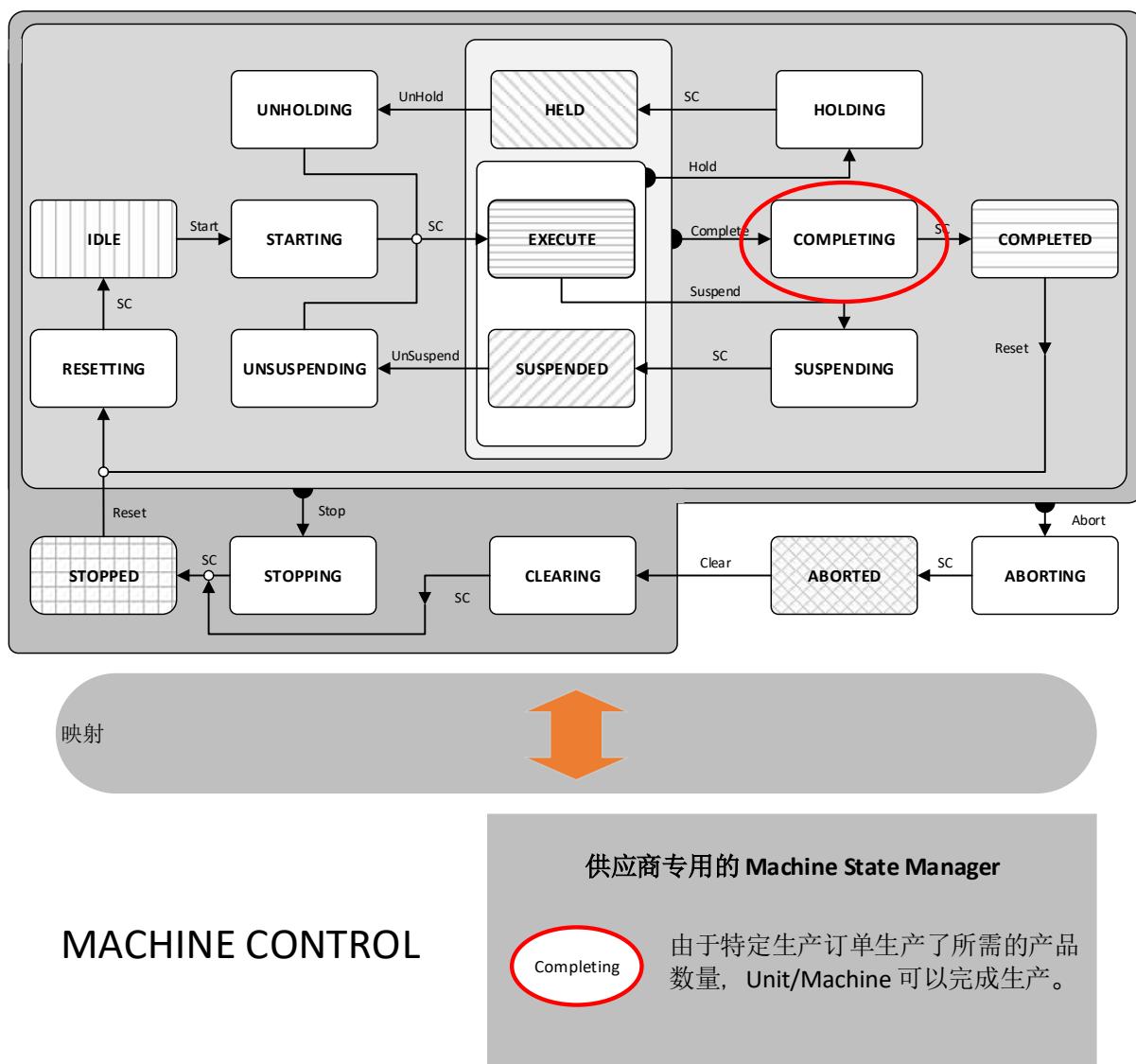


图 30：PackML Interface State Model 和 Machine control State model 之间的 Completing 映射关系

7.9 STOPPING

STOPPING 状态的目的是从监控控制系统停止生产订单的生产，或在操作员希望停止生产时在 Unit/Machine 上停止生产订单的生产。

表 11: STOPPING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
STOPPING	<p>此状态会执行控制逻辑，让 Unit/Machine 像 STOPPED 状态反映的那样受控停机。</p> <p>Unit/Machine 可通过外部接口提供机器参数。机器参数可用于了解执行期间在 Unit/Machine 上手动更改和更正的信息。</p> <p>机器供应商和最终用户必须指定机器进入 STOPPING 和 STOPPED 状态时要发生的特殊状况。例如，Unit/Machine 应该清空内部产品缓冲区，还是应该立即停机。</p>	Stopping	<p>此状态是响应 <i>STOP</i> 命令而进入的。在此状态下，机器会执行逻辑，让 Unit/Machine 像 STOPPED 状态反映的那样受控停机。</p> <p>除非执行了 <i>RESETTING</i>，否则无法正常启动机器。</p> <p>完成受控停机请求的结果是，Unit/Machine 将进入 STOPPED 状态。</p>

PACKML INTERFACE (STOPPING 源自 stopping)

由于操作员停止了生产订单/批次，生产订单/批次的执行 STOPPING。

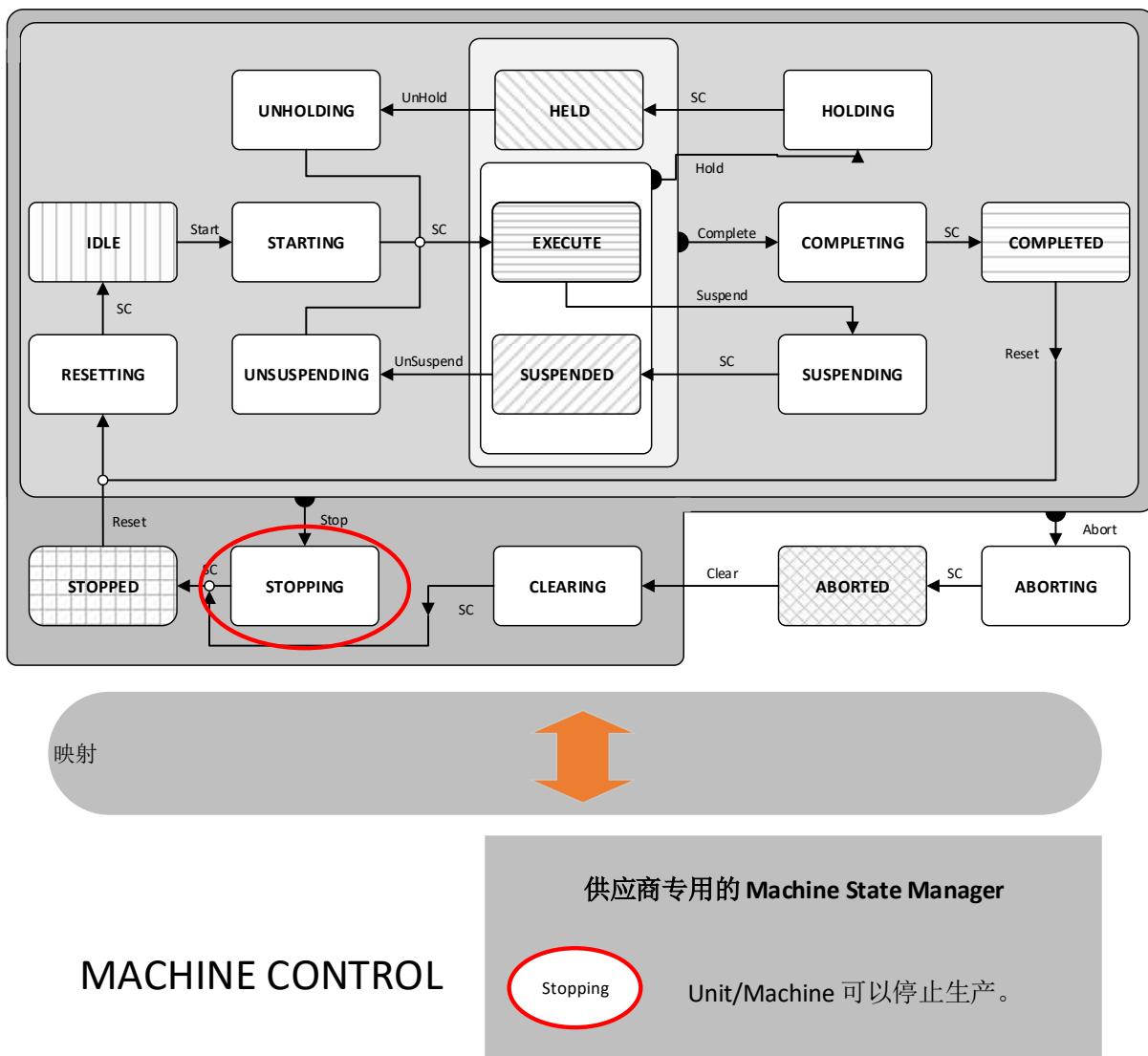


图 31: PackML Interface State Model 和 Machine control State model 之间的 Stopping 映射关系

7.10 ABORTING

表 12: ABORTING 状态解释

PackML Interface State	PackML interface 说明 (PackML Interface State Manager)	Machine State	Machine State 说明 (供应商专用 Machine State Manager)
ABORTING	<p>响应 Abort 命令时可随时进入 ABORTING 状态。Aborting 逻辑会令 Unit/Machine 迅速地安全停机。 Aborting 逻辑会提供必要的动作，将 Unit/Machine 切换到 ABORTED 状态。 Unit/Machine 进入 Aborting 状态时要执行的操作由机器供应商和最终用户定义。</p>	Aborting ¹⁹	<p>响应 ABORT 命令时或发生机器故障时会随时进入 ABORTING 状态。 ABORTING 逻辑会令机器迅速地安全停机。 ABORTING 是按下紧急停机按钮或需要快速停止 Unit/Machine 的机器故障造成的。</p>
		Stopping	<p>此状态是响应 STOP 命令而进入的。在此状态下，机器会执行逻辑，让 Unit/Machine 像 stopped 状态反映的那样受控停机。除非执行了 PackML RESETTING，否则无法正常启动机器。</p>

¹⁹在 Aborting 前生产的产品处于包装阶段，因此通常不会受损。Aborted 状态表示生产订单已完成，无法重新启动。

PACKML INTERFACE (ABORTING 源自 stopping)

由于操作员终止了生产订单，生产订单的执行 ABORTED。有两种不同方法可映射到 Abort state:

- 1) Unit/Machine 将 Stopped 状态映射到 PackML Aborted state, 或
- 2) Unit/Machine 将 Aborted 状态映射到 PackML Aborted state。

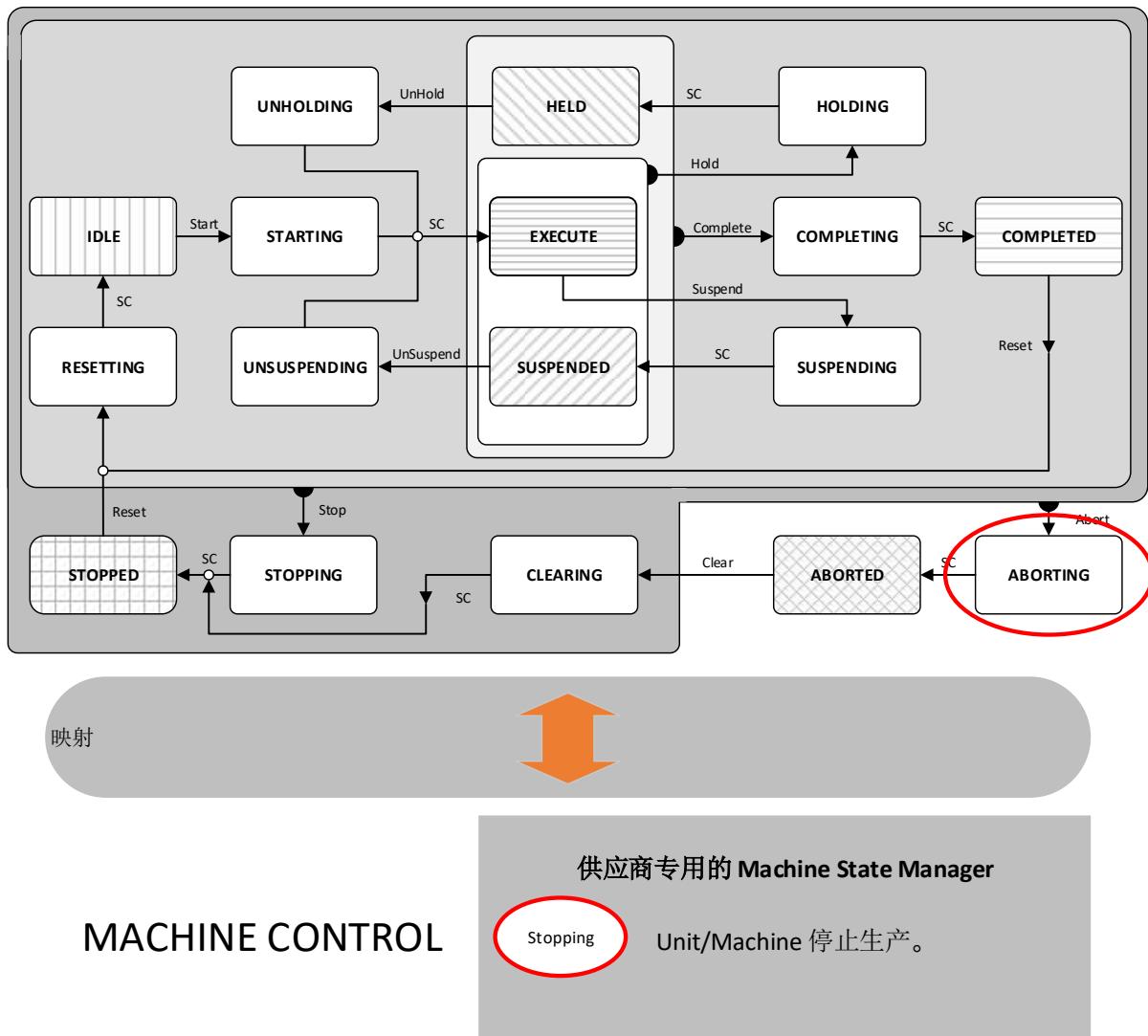


图 32: PackML Interface State Model 和 Machine control State model 之间的 Stopping 映射关系

PACKML INTERFACE (ABORTING 源自 aborting)

由于操作员终止了生产订单，生产订单的执行 ABORTED。有两种不同方法可映射到 Abort state:

- 1) Unit/Machine 将 Stopped 状态映射到 PackML Aborted state, 或
- 2) Unit/Machine 将 Aborted 状态映射到 PackML Aborted state。

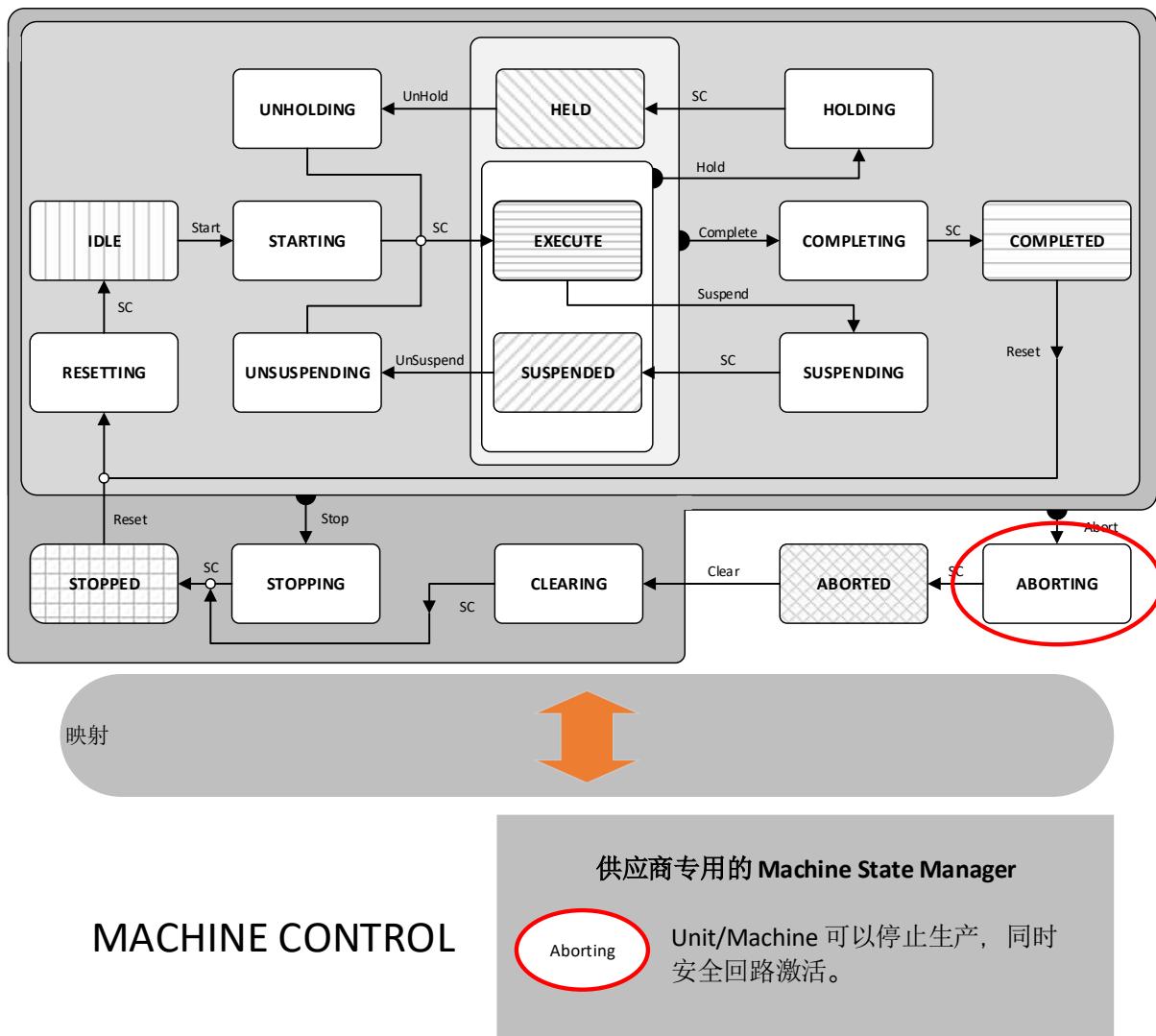


图 33: PackML Interface State Model 和 Machine control State model 之间的 Abort 映射关系

7.11 MAPPING TABLE

如下图所示，表 13 概要列出了 PackML Interface State 和一组供应商专用 Machine State 之间的各种映射可能性的示例。供应商专用 Machine State Model 的示例在图 34 中进行了描述。

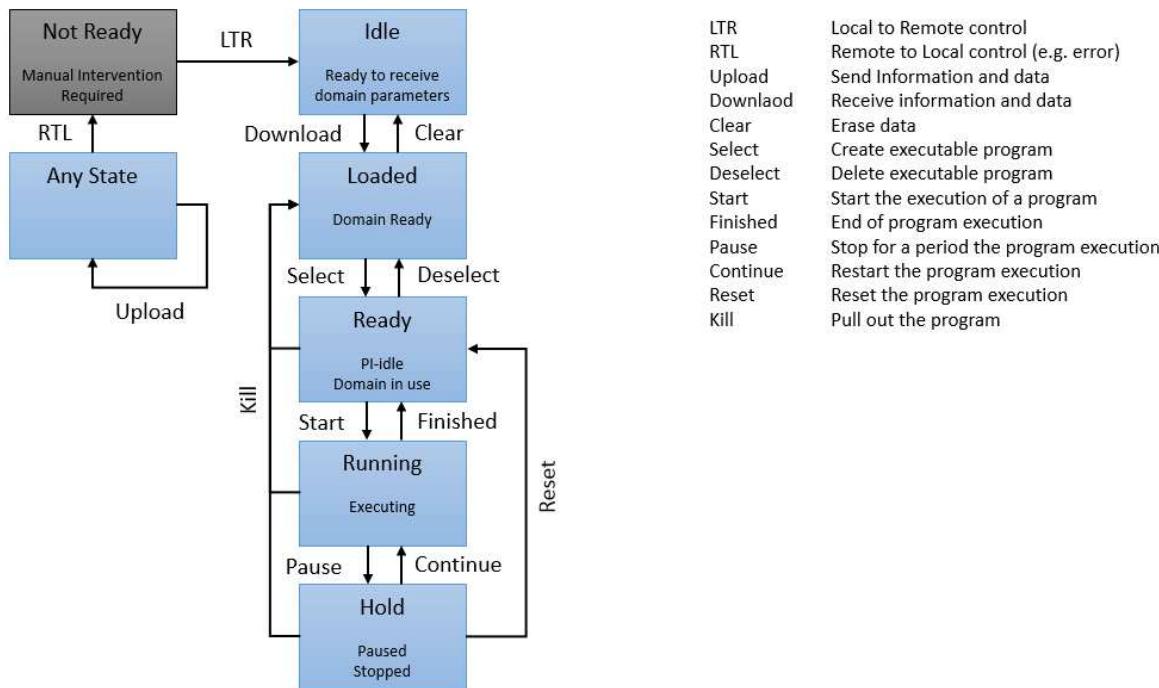


图 34：供应商专用 Machine State model 示例

该映射处理决定了哪些供应商专用 Machine State 和命令与哪些 PackML State 和命令对应。上面的状态模型显示了可映射到 PackML acting state (PackML state model diagram 中的白框) 的若干状态和大量状态事务，如表 13 所示。

表 13：PackML Interface State 和 Machine State 之间的映射表示例

Supplier dependent Machine State & transitions																										
PackML Interface State	Not ready (Manual intervention required)		LTR – Local to Remote control		RTL – Remote to local control (E.g. error)		Download (Not supported by PackML)		Select (Not supported by PackML)		Deselect (Not supported by PackML)		Ready													
	STOPPED	X	RESETTING	C	IDLE	X	STARTING	C	EXECUTE	X	HOLDING	C	HELD	X	UNHOLDING	C	SUSPENDING	SUSPENDED	UNSUSPENDING	COMPLETING	COMPLETE	ABORTING	ABORTED	CLEARING	C	STOPPING
STOPPED	X																									
RESETTING		C																								
IDLE			X			X																				
STARTING															X C											
EXECUTE																			X							
HOLDING				C																						
HELD	X																									
UNHOLDING		C																		C						
SUSPENDING																				C						
SUSPENDED																				X						
UNSUSPENDING																			C							
COMPLETING																	C									
COMPLETE																		X								
ABORTING				C																						C
ABORTED	X																									
CLEARING		C																								
STOPPING			C																	C						

X = State, C = Command (Acting state)

提示：机器供应商和最终用户需要就机器何时进入不同的 PackML interface state 达成一致。建议制作一个如表 13 那样的对照表，将 PackML Interface State 作为该表格的一个维度，将当前的机器专用 Machine State 作为表格的另一个维度。同时，列出不同 PackML Interface state 下所有可能的情况。

提示：在规定具体的 Unit/Machine 的过程中，使用表 13，并填写具体的供应商专用 Machine State 和转换。

7.12 MAPPING ALARMS TO EVENT TRIGGERS TO PACKML

我们可以制作一个对照表，以方便机器供应商配置哪些类型的报警和事件会导致 PackML State 变化。可以用来指定某个机器何时应该切换到 PackML state model 中的状态。参见表 14 作为映射表示例。

而且，我们要特别关注当 Unit/Machine 需要跳转至 HELD、STOPPED、SUSPENDED 和 ABORTED 状态时的情况，因为最终用户有时对何时应进入不同的状态有不同的解释和定义。

表 14：与 PackML 状态管理表对应的错误和事件触发器

Events / Alarms	State change ID = Commands	Reaction for Machine Control	Reaction for PackML Interface state manager
1	8	Abort	ABORTING
2	8	Abort	ABORTING
3	4	Hold – Error on material	HOLDING
4	4	Hold – Error on glue	HOLDING
5	3	Stop – Drive failure	ABORTING
6	3	Stop – Sensor failure	HOLDING
7	6	Suspend - Blocked	SUSPENDING
8	6	Suspend – Queue	SUSPENDING
9	20	Pop-up information on HMI – Executing state	No state change
10	20	Pop-up information on HMI – Executing state	No state change
...

提示：在规定具体的 Unit/Machine 的过程中，建议使用未记录的表 14，并填写具体的供应商专用机器报警和警告以及 PackML Interface State Manager 的动作。

8. PACKML CONTROL COMMAND DEFINITIONS

控制命令会提供状态命令，触发 PackML Interface State Model 中的状态变化（请参见图 10：PackML Interface State Model）。在 Unit/Machine 控制系统中，此命令的处理可与远程或本地 Unit/Machine 条件结合使用，从而触发接口状态模型从 Wait state 切换到 Acting state。Unit/Machine 应始终响应命令²⁰，但不是所有命令都能造成 PackML state 变更。

8.1 PACKML COMMANDS

控制命令可由 Unit/Machine HMI 设置或通过 PackML interface 远程设置。表 15 的“Issued”列，说明了谁可以发出命令。

控制命令应为整数类型。请参见第 8.2 节了解特定的命令值。

表 15：控制内部、外部和紧急停机命令

命令	Issued			Comments
	On or by unit/ machine ²¹	By External interface	E-stop	
Start	YES (*) 	YES		<p>Start 命令会将 Unit/Machine 切换到 Starting 状态。</p> <p>Start 命令最终会将 Unit/Machine 置于 EXECUTE 状态。如果出现错误，Unit/Machine 会被置于 STOPPED 或 ABORTED 状态。Unit/Machine 处于生产模式和 IDLE 状态时，允许外部系统发出 Start 命令。从外部系统启动 Unit/Machine 时，一旦在 STOPPED 或 IDLE 状态下下载了参数，即会自动启动 Unit/Machine，因为操作员已在本地批准生产（请参见 reset 命令）。</p> <p>(*) 如果并无外部系统，可在 Unit/Machine 上进行启动。</p>
Hold	YES			<p>Hold 命令可在检测到内部 Unit/Machine 故障时由 Unit/Machine 执行，或在生产期间由操作员在 Unit/Machine 上执行。</p> <p>Hold 命令会将 Unit/Machine 从 EXECUTE 状态切换到 HOLDING 状态。</p> <p>利用 Hold 命令，操作员可在工序中安全地执行手动干预，例如移除进料区中破损的物料。或打开安全门移走纸板。本指南中并未介绍安全停机的配置。</p>

²⁰ 如果某个命令对当前状态无效，Unit/Machine 不应做出响应，而是应该发出警告。例如，向处于 STOPPED 状态的机器发送 start 命令时，Unit/Machine 应该作出拒绝该命令的响应。

²¹ “Issued On or by unit/ machine” 列指出操作员可通过 HMI 或通过机器上的硬件按钮发出命令。该列也指出机器本身也可发出命令。例如，Unsuspend 命令并不需要由操作员发出。

命令	Issued			Comments
	On or by unit/ machine ²¹	By External interface	E-stop	
Unhold	YES 			<p>Unhold 命令会解除 Hold 状态，将 Unit/Machine 切换至 UNHOLDING 状态，然后再恢复至 EXECUTE 状态。</p> <p>Unhold 命令是对操作员在机器出现报警或警告指示的错误/故障后发出的恢复生产命令的响应。例如，当操作员清除了堵塞的物料时。</p> <p>Unhold 命令始终需要由操作员发出，绝不会由机器自动发出。</p>
Suspend	YES	YES		<p>来自 Unit/Machine 的 Suspend 命令是进料区缺料，或出料区物料饱和以致于 Unit/Machine 无法继续稳定生产，又或根据操作员命令的结果。</p> <p>Unit/Machine 已处于 SUSPENDED 状态时，如果收到外部 Suspend 命令，则只会更改停机原因。</p> <p>Suspend 命令可能是 Unit/Machine 以外的上游或下游单元出现问题造成的结果。</p>
Unsuspend	YES	YES		<p>Unsuspend 命令会解除 Unit/Machine 的 SUSPENDED 状态，将 Unit/Machine 切换至 UNSUSPENDING 状态，然后再恢复至 EXECUTE 状态。当下游和上游机器做好执行准备，或当外部系统发出 Unsuspend 时，机器会发出 Unsuspend 命令。</p>
State Completed (SC) from any acting states	YES			<p>State Completed (SC) 是一个内部 Unit/Machine 动作/命令，它会将 Unit/Machine 本身切换到下一个状态。SC 在适当的 Unit/Machine 条件满足时触发。</p>
Complete	YES	YES		<p>完整的命令可以基于 Unit/Machine 条件在内部生成，例如达到预定义的生产计数，或者在外部生成。这意味着生产订单已完成。Complete 命令将 Unit/Machine 从 EXECUTE、HELD 或 SUSPENDED 状态切换到 COMPLETING 状态，最终达到 COMPLETE 状态。</p> <p>提示：通过外部接口可使用 Stop 命令强制 Unit/Machine 完成订单，而且必要时还可清空 Unit/Machine 的产品输出缓冲区。</p>

命令	Issued			Comments
	On or by unit/ machine ²¹	By External interface	E-stop	
Stop	YES 	YES		<p>Stop 命令会执行控制逻辑，将 Unit/Machine 置于 STOPPING 状态，最终再置于 STOPPED 状态。</p> <p>操作员可通过外部接口或在 Unit/Machine HMI 上停止 Unit/Machine。Unit/Machine 将停止，并通过 PackML interface 提供参数数据。</p> <p>Stop 命令可在除 ABORTED、ABORTING、CLEARING、STOPPING 和 STOPPED 以外的任何状态下发出。这意味着，Stop 命令可在 PackML interface State Model（请参见图 10: PackML Interface State Model）的灰色区域内发出。</p> <p>若要完成生产，可在 Unit/Machine 处于 EXECUTE、HELD、SUSPENDED、COMPLETE 或 IDLE 状态下且 Unit/Machine 处于生产模式时向机器发出外部 Stop 命令。</p>
Reset ²²	YES 			<p>Reset 命令由操作员在 Unit/Machine HMI 上触发。</p> <p>操作员需要在做好生产准备的 Unit/Machine 上触发，并将 Unit/Machine 经由 RESETTING 状态置于 IDLE 状态。</p> <p>操作员需要确认所有安全条件都已满足，而且 Unit/Machine 做好了生产准备。</p> <p>处理 Reset command 时，Unit/Machine 可处于 STOPPED 或 COMPLETE 状态。</p>
Abort			YES 	<p>Abort 命令将在机器上触发紧急停机时由机器自动处理。</p> <p>Abort 命令可在任何状态下执行（请参见图 10 – 浅灰色和深灰色），而且会将 Unit/Machine 切换到 Aborting 状态，ABORTING 转换状态以及 ABORTED 安全状态。</p> <p>紧急停机操作会导致 Unit/Machine 的安全系统触发，并发送 Abort 命令。此时采取何种技术解决方案来确保 Unit/Machine 切换至并保持在 ABORTED 状态下仍待确定。</p>
Clear	YES			Unit/Machine 被 Abort 命令停止后使用 Clear 命令。Clear 命令在 Unit/Machine 的 HMI 上触发，用来清除 ABORTING 时可能会出现以及 ABORTED 状态下可能存在的错误，然后进入 CLEARING 状态，再进入 STOPPED 状态。

²² Reset 命令不可或缺，它可确保 Unit/Machine 能从远程安全启动。

8.2 UNIT HMI 上和来自外部系统的命令的规范

机器供应商应该填写下表，确定哪些命令可从 Unit HMI 触发，哪些可通过接口从外部系统触发。

表 16: Command 的定义

Command	PackML Tag Value	Implemented on Unit			
		External interface	Unit push buttons and/or HMI action	E-stop	Name on unit HMI
No command	0	Optional	N/A		N/A
Reset	1	Optional	YES		Reset
Start	2	Optional	YES		Start
Stop	3	Optional	YES		Stop
Hold	4	Optional	YES		N/A
Unhold	5	Optional	YES		Restart
Suspend	6	YES	YES		N/A
Unsuspend	7	YES	YES		N/A
Abort	8	Optional		YES	E-Stop
Clear	9	Optional	YES		Reset E-Stop
Complete	10	Optional	YES		End or Stop Order

提示: 在规定具体的 *Unit/Machine* 的过程中，建议使用未记录的表 16，并填写具体的供应商专用 *Machine command* 和同等 *HMI* 名称。

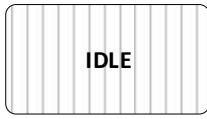
9. PACKML INTERFACE STATE 定义

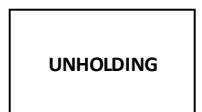
Interface Unit/Machine State 定义机器的当前状况。Interface Unit/Machine State 是指通过 HMI 或外部控制系统从总体角度看到的 Unit/Machine 的总体状态。Unit/Machine 可能拥有若干内部控制器（PLC 或驱动控制器），但从 PackML 角度来看，Unit/Machine 将被视为一个单元。

状态为整数值，如表 17 所示。

Unit/Machine 通电后，它首先处于 STOPPED 状态。

表 17: Interface Unit/Machine State 定义

状态名称	PackML Tag Value	说明
IDLE 	1	<p>此状态表示 Resetting 已完成。Unit/Machine 已准备好开始新的生产，而且收到了机器参数。如果机器上出现错误，Unit/Machine 将进入 STOPPING 状态并止于 STOPPED 状态。</p> <p>在 IDLE 状态下，Unit/Machine 满足了所有安全条件，并准备好进行生产。在 IDLE 状态下，Unit/Machine 允许来自外部系统的远程 Start 命令。最终用户应负责确保外部 Start 命令符合公司安全要求。</p>
STARTING 	3	<p>此状态提供启动 Unit/Machine 所需的步骤，而且是 Start 命令的结果。该启动逻辑会将 Unit/Machine 参数加载到机器控制器中，并在这些参数妥善加载后，将 Unit/Machine 切换到 EXECUTE 状态。该启动逻辑将逐渐提升速度、压力等，从而为生产做好准备。但是，如果加载的数据存在错误，该启动逻辑会将 Unit/Machine 切换到 STOPPING 状态，随后 Unit/Machine 将止于 STOPPED 状态。除非收到 Start 命令，否则不允许 Unit/Machine 切换到 EXECUTE 状态。</p> <p>如果机器上出现错误，Unit/Machine 将进入 STOPPING 状态并止于 STOPPED 状态。</p> <p>在 STARTING 状态下要执行的动作须由机器供应商和最终用户指定。</p>
EXECUTE 	6	<p>Unit/Machine 生产物料后，它即处于 EXECUTE 状态。</p> <p>机器和 Unit/Machine 正在运行，而且所选模式和/或所选配方定义的所有条件都满足。</p>

状态名称	PackML Tag Value	说明
COMPLETING 	16	<p>此状态会提供必要的步骤来完成任务，例如在机器中的产品计数器达到指定数量的结果。操作员在 Unit/Machine 上更正的参数（配方控制参数）可提供给外部系统。机器参数可通过外部系统使用，从而了解执行期间在 Unit/Machine 上手动更改或更正的数据信息。</p> <p>机器供应商和最终用户需要指定在 Unit/Machine 进入 COMPLETING 和 COMPLETE 状态时要发生的特殊状况。例如，Unit/Machine 应该清空内部产品缓冲区，还是应该在机器中仍有产品时立即停机。</p> <p>在 COMPLETING 状态下要执行的动作须由机器供应商和最终用户指定。</p>
COMPLETE 	17	<p>Unit/Machine 已完成了 COMPLETING 状态，现在正等待 Reset 命令，以便切换到 Resetting 状态。Unit/Machine 通电，且所有数据仍保存在机器中。Unit/Machine 已准备好复位。</p>
Resetting 	15	<p>RESETTING 状态将启动复位程序，这通常会导致 Unit/Machine 清除数据并将 Unit/Machine 置于 IDLE 状态。在 IDLE 状态下，Unit/Machine 组件处于通电状态下并等待 Start 命令。将机器复位后，数据（配方和参数）会被清除，此时 Unit/Machine 将准备接收新的机器参数。例如，操作员可能已更换工具，而通过激活 Resetting 程序，操作员表明机器做好了生产准备。</p> <p>在 Resetting 状态下要执行的动作须由机器供应商和最终用户指定。</p>
HOLDING 	10	<p>HOLDING 逻辑会令 Unit/Machine 受控停机，或者会将特定单元置于代表 HELD 的状态。</p> <p>这是 Unit/Machine 上出现错误/故障时或操作员从 Unit/Machine HMI 启动 Hold 命令时执行的内部控制逻辑。Holding 控制逻辑会将 Unit/Machine 从 HOLDING 状态切换为 HELD 状态。</p> <p>紧急停机属于异常，它需要单独处理，而不属于 Holding 逻辑的一部分。请参见 Aborting 和 ABORTED 状态。</p>
HELD 	11	<p>HELD 状态会将机器操作挂起，以便在此期间清除物料堵塞或安全地更正设备故障，然后才会恢复生产。</p> <p>为了能在 HELD 状态后妥善地恢复生产，执行 HOLDING 程序时，必须将收到 HOLD 命令时所有相关工序设置点和流程恢复状态信息保存到 Unit/Machine 控制器中。</p>
UNHOLDING 	12	<p>UNHOLDING 状态是对操作员执行的恢复生产命令的响应。</p> <p>UNHOLDING 状态下的控制逻辑会让 Unit/Machine 做好重新进入正常 EXECUTE 状态的准备。该 Unholding 逻辑将逐渐提升速度、压力等，从而为生产做好准备。</p>

状态名称	PackML Tag Value	说明
SUSPENDING 	13	SUSPENDING 状态是进入 SUSPENDED 等待状态前必须经过的一个状态，它会让 Unit/Machine 为停止熨烫、停止切割、停止除袋等操作做好准备。此时 Unit/Machine 可能会在相关的设置速度下运行，但并没有为下游的下一个 Unit/Machine 生产产品。执行受控的 Suspending 程序期间，Unit/Machine 将切换至 SUSPENDED 状态。 机器供应商和最终用户必须指定机器进入 SUSPENDING 和 SUSPENDED 状态时要发生的特殊状况。例如，Unit/Machine 应该装满内部产品缓冲区，还是应该立即停机。 Suspending 是进料区缺料或出料区物料饱和，以致于 Unit/Machine 无法 Executing 继续稳定的生产。当操作员由于外部事件需要休息或暂停时，操作员可通过外部接口强制触发 SUSPENDING 状态。
SUSPENDED 	5	SUSPENDED 状态可由外部工序条件触发。 在 SUSPENDED 状态下，Unit/Machine 会等待外部工序条件恢复正常，然后会切换至 UNSUSPENDING 状态，继而是正常的 EXECUTE 状态。
UNSUSPENDING 	14	此状态是 Unit/Machine 在 SUSPENDED 状态下生成返回 EXECUTE 状态的请求或通过外部接口发出 Unsuspend 命令的结果。 此状态在 EXECUTE 状态之前完成，并会让 Unit/Machine 做好进入 EXECUTE 状态的准备。 此状态下的动作可能包括提升速度、打开真空以及重新吸合离合器。 Unit/Machine 进入 UNSUSPENDING 状态时要执行的操作由机器供应商和最终用户定义。
Stopping 	7	此状态会执行控制逻辑，让 Unit/Machine 像 STOPPED 状态反映的那样受控停机。 机器供应商和最终用户必须指定 Unit/Machine 进入 Stopping 和 STOPPED 状态时要发生的特殊状况。例如，Unit/Machine 应该清空内部产品缓冲区，还是应该立即停机。
STOPPED 	2	完成 STOPPING 状态后，Unit/Machine 通电，但无动作。此时与其他系统的所有通信仍正常进行。Unit/Machine 通电，且所有参数仍保存在机器中。Unit/Machine 已准备好复位。 除非执行了 Resetting，否则无法正常启动 Unit/Machine。
Aborting 	8	响应 Abort 命令时可随时进入 ABORTING 状态。Aborting 逻辑会令 Unit/Machine 迅速地安全停机。Aborting 逻辑会提供必要的动作，将 Unit/Machine 切换到 ABORTED 状态。 Unit/Machine 进入 ABORTING 状态时要执行的操作由机器供应商和最终用户定义。

状态名称	PackML Tag Value	说明
ABORTED 	9	处于此状态的 Unit/Machine 需要维持 Unit/Machine 状态信息。Unit/Machine 只能在明确的 Clear 命令后退出 ABORTED 状态，再通过手动干预更正和复位检测到的安全回路，例如急停。
Clearing 	1	在 CLEARING 状态下，操作员需要清除安全系统，将 Unit/Machine 切换到 STOPPED 状态。Unit/Machine 清除 Aborting 状态时发生的错误，进入 ABORTED 状态，然后切换至 STOPPED 状态。 Unit/Machine 会将机器参数提供给外部系统。机器参数可用于了解 Unit 上最终手动更改和更正的数据信息。 Unit/Machine 进入 CLEARING 状态时要执行的操作由机器供应商和最终用户定义。

10. UNIT CONTROL MODES

Unit/Machine 可能位于不同模式下，例如 Production、Maintenance、Manual、Clean、Jog Mode 等。

Unit control mode 是一个有序的状态和命令子集，用于确定 Unit/Machine 工序执行的策略。

这意味着上面的 PackML Interface State Model 在不同模式下可能有不同的呈现方式。PackML Interface State Model 必须在 Production mode 下全部实施，在 Manual mode 和 Maintenance mode 下部分实施。在 Manual mode 下，与操控 Unit/Machine 的外部系统的所有通信都会被禁用。在 Manual mode 下，操作员无需连接至外部系统，只需通过 Unit HMI 就可接管对机器的控制。

机器供应商必须指定除了 Production mode、Manual mode 和 Maintenance mode 外还有哪些模式可以使用。

Unit Control mode 的数据类型为整数，详见表 18 所定义。

机器供应商和最终用户可以定义开机后的启动模式，例如 STOPPED 或 IDLE。

表 18：必需（Mandatory）和可选（Optional）的 Unit/Machine 模式

PackML Tag Value	Mode	PackML State Model	Restricted Access	用途
1	PRODUCTION	Mandatory	NO	此模式为用于所有生产活动的主要模式，在该模式下会用到所有 PackML model state。PackTag 处于活动状态且报告数据。Unit/Machine 在响应主要来自外部系统或直接由操作员输入的命令时执行相关逻辑。
2	MAINTENANCE	Optional	YES	用于例行的预防性 Maintenance 或计划内 Maintenance。应该用于记录计划内 maintenance 操作及维护持续时间。 此模式可供授权人员独立于其他系统单独运行 Unit/Machine。此模式通常用于故障排查、机器试验或测试操作改进。 例如，清洁打印头就属于 maintenance 任务。
3	MANUAL	Optional	YES	用于计划外技术干预的故障诊断。 <i>提示：</i> 在 Manual Control mode 下，没有必要实施所有 PackML state。

用户自定义的 PackML Mode 示例

Unit/Machine 上也可能会提供更多的 Unit Control Mode，但机器无需遵循这些其他 Unit Control Mode 中的 PackML Interface State Model。

PackML Tag Value	Mode	PackML State Model	Restricted Access	用途
4	CHANGE OVER	Optional	NO	可专用于格式或配方更改，包括“cleaning”操作。
5	CLEAN	Optional	NO	可用于 Routine Cleaning 要求，例如工厂 maintenance 程序规定的清洁要求。
6	SET UP	Optional	Optional restriction	可用于设置或调整，例如机械调整和测试
7	EMPTY OUT ²³	Optional	NO	将用于清空机器，例如在工厂周末停机前的批次生产结束后，清空机器内驻留的制成品，最大限度减少产品“rework”数量。
To be defined	OEM specific	Optional	YES	仅用于 OEM，此机器所需的所有设置很可能不是任何客户所需要的。
To be defined	?	Optional	?	?

提示：模式更改²⁴只能在 Unit/Machine 处于 STOPPED 状态下进行。这意味着退出和进入某个模式都必须在 STOPPED 状态下。模式更改不会对 PackML state 作出任何更改。例如，处于 Production mode 和 Stopped 状态下的 Unit/Machine 将进入 STOPPED 状态下的 Maintenance mode。模式更改也可发生在 ABORTED 状态下，即在该状态下进入 Manual mode。但是，出于安全考虑，退出 Manual mode 会受到一定限制。

²³ 切换到 mode EMPTY OUT 的 mode change 可能要求在 EXECUTE 状态下执行模式更改。此模式更改可由操作员手动执行，也可由 Unit/Machine 在生产订单结束时自动执行。

²⁴ 根据 ISA-TR88.00.02-2022，允许在所有 PackML 等待状态下更改模式。

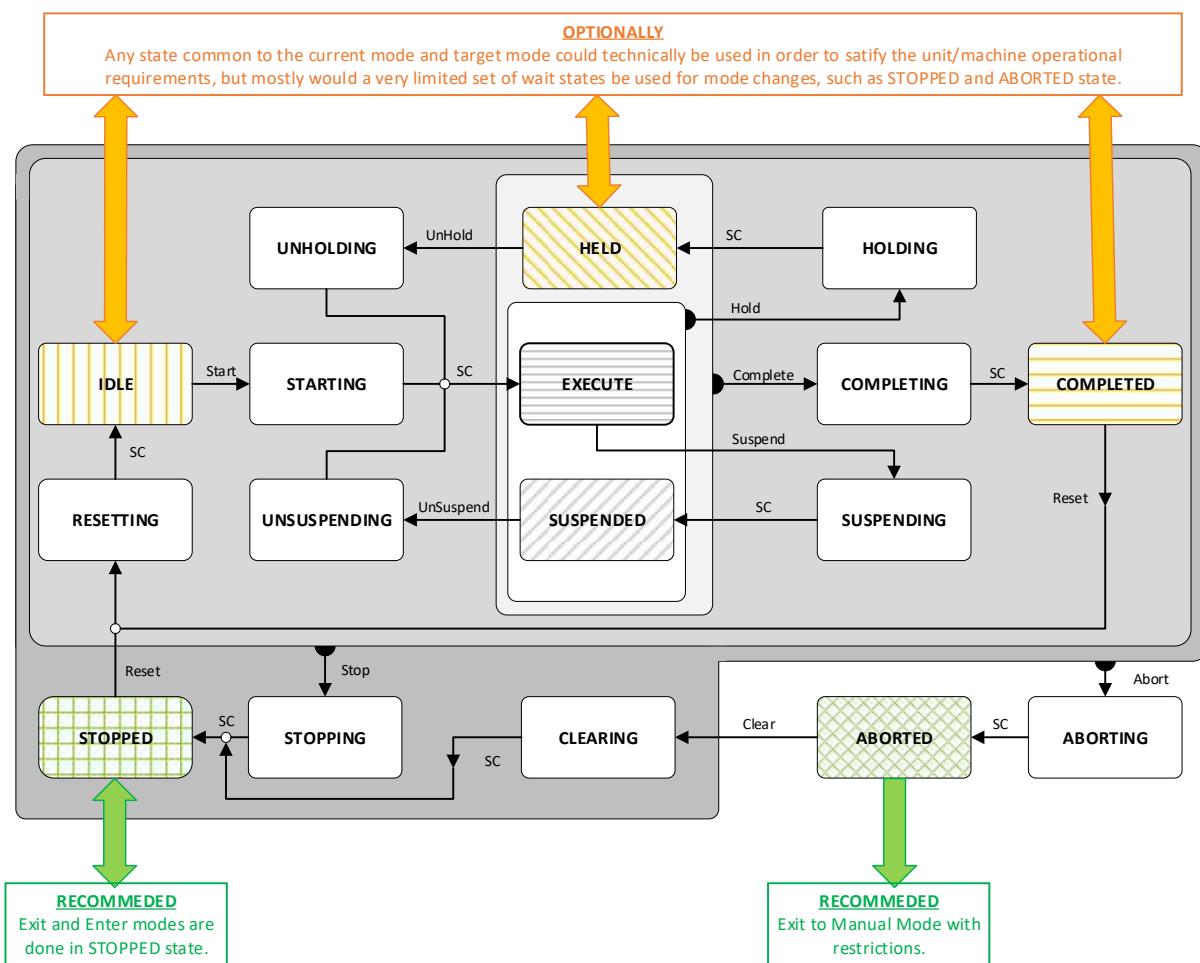


图 35：可以变更模式的推荐 PackML State

11. PACKTAG

PackTag 为基本状态模型中使用的数据元素提供了一套统一的命名约定。PackTag 是机械工业化行业为了数据交换而在开放架构中使用的命名数据元素。本文包含最终用户最小数据元素及其数据类型，详见表 19 所定义。PackTag 对机器之间的通信乃至机器与更高级信息系统之间的数据交换很有用。

对 PackTags 进行了重大更改，删除了“远程”界面，并将通过“配方”描述 Unit/Machine 配置的 PackTags 添加到监控系统。

相比 PackML Guideline 第一版，增加了“远程”界面，并将通过“配方”描述 Unit/Machine 配置的 PackTags 添加到监控系统。参数已分为下述变量类型：REAL、STRING、LREAL 和 DINT。

提示：请使用 *Pascal Case*，即标识符中每个单词的首字母大写，以方便阅读。为确保与所有系统的互操作性，IEC 61131 并不区分大小写，因此建议遵循 *Pascal Case* 命名法。此外，用户也可使用下划线（“_”）替代不支持结构化标签名称的旧系统中使用的点号。

下表说明了最终用户建议机器供应商实施的最小 PackTag 集。当然，机器供应商也可实施额外的 PackTag。所需的最小 PackTag 集位于“End user Minimum tags”列中。

表 19：最小 STATUS PackTag²⁵

PackTag type	PackTag	Example of End user term	Datatype IEC 61131-3	TR 88.00.02 Minimum tags	End user Minimum tags
Status	StateCurrent	State	DINT	X	X
Status	UnitModeCurrent	Mode	DINT	X	X
Status	MachSpeed	Nominal Speed	REAL	X	X
Status	CurMachSpeed	Current Speed	REAL	X	X
Status	EquipmentInterlock.Blocked	Blockage	BOOL	X	X
Status	EquipmentInterlock.Starved	Starvation	BOOL	X	X
Status	Parameter_REAL [#]	Machine data/parameter	Array Structure		X
Status	Parameter_REAL [#].ID	Parameter ID	DINT		X
Status	Parameter_REAL [#].Name	Name of parameter	STRING		X
Status	Parameter_REAL [#].Unit	Unit of measure	STRING[5]		X
Status	Parameter_REAL [#].Value	Value of parameter	REAL		X
Status	Parameter_STRING [#]	Machine data/parameter	Array Structure		X
Status	Parameter_STRING [#].ID	Parameter ID	DINT		X
Status	Parameter_STRING [#].Name	Name of parameter	STRING		X
Status	Parameter_STRING [#].Unit	Unit of measure	STRING[5]		X
Status	Parameter_STRING [#].Value	Value of parameter	STRING		X
Status	Parameter_LREAL [#]	Machine data/parameter	Array Structure		X
Status	Parameter_LREAL [#].ID	Parameter ID	DINT		X
Status	Parameter_LREAL [#].Name	Name of parameter	STRING		X
Status	Parameter_LREAL [#].Unit	Unit of measure	STRING[5]		X
Status	Parameter_LREAL [#].Value	Value of parameter	LREAL		X
Status	Parameter_DINT [#]	Machine data/parameter	Array Structure		X
Status	Parameter_DINT [#].ID	Parameter ID	DINT		X
Status	Parameter_DINT [#].Name	Name of parameter	STRING		X
Status	Parameter_DINT [#].Unit	Unit of measure	STRING[5]		X
Status	Parameter_DINT [#].Value	Value of parameter	DINT		X
Status	RecipeCurrent	Number of Recipe	DINT		X
Status	RecipeRequested	Requested Recipe No	DINT		X

²⁵ ANSI/ISA-TR88.00.02-2022 是所有 PackTag 的参考。

PackTag type	PackTag	Example of End user term	Datatype IEC 61131-3	TR 88.00.02 Minimum tags	End user Minimum tags
Status	RecipeChangeInProcess	Recipe is changing	BOOL		
Status	Recipe ²⁶ [#]	Recipe structure	Array Structure		X
Status	Recipe [#].ID	Recipe ID	DINT		X
Status	Recipe [#].Name	Recipe Name	STRING		X
Status	Recipe [#].Unit	Unit of Measure (E.g., Pc, Crtn, Case, ...)	STRING[5]		X
Status	Recipe [#].PrimaryQty	Primary	REAL		X
Status	Recipe [#].ProcessVariables ²⁷ .	Process variables			X
Status	Recipe [#].ProcessVariables.Parameter_REAL [#]	Process variables Machine data/parameter	Array Structure		X
Status	Recipe [#].ProcessVariables.Parameter_REAL [#].ID	Process variables Parameter ID	DINT		X
Status	Recipe [#].ProcessVariables.Parameter_REAL [#].Name	Process variables Name of parameter	STRING		X
Status	Recipe [#].ProcessVariables.Parameter_REAL [#].Unit	Process variables Unit of measure	STRING[5]		X
Status	Recipe [#].ProcessVariables.Parameter_REAL [#].Value	Process variables Value of parameter	REAL		X
Status	Recipe [#].ProcessVariables.Parameter_STRING [#]	Process variables Machine data/parameter	Array Structure		X
Status	Recipe [#].ProcessVariables.Parameter_STRING [#].ID	Process variables Parameter ID	DINT		X
Status	Recipe [#].ProcessVariables.Parameter_STRING [#].Name	Process variables Name of parameter	STRING		X
Status	Recipe [#].ProcessVariables.Parameter_STRING [#].Unit	Process variables Unit of measure	STRING[5]		X
Status	Recipe [#].ProcessVariables.Parameter_STRING [#].Value	Process variables Value of parameter	STRING		X
Status	Recipe [#].ProcessVariables.Parameter_LREAL [#]	Process variables Machine data/parameter	Array Structure		X
Status	Recipe [#].ProcessVariables.Parameter_LREAL [#].ID	Process variables Parameter ID	DINT		X
Status	Recipe [#].ProcessVariables.Parameter_LREAL [#].Name	Process variables Name of parameter	STRING		X
Status	Recipe [#].ProcessVariables.Parameter_LREAL [#].Unit	Process variables Unit of measure	STRING[5]		X
Status	Recipe [#].ProcessVariables.Parameter_LREAL [#].Value	Process variables Value of parameter	LREAL		X
Status	Recipe [#].ProcessVariables.Parameter_DINT [#]	Process variables Machine data/parameter	Array Structure		X
Status	Recipe [#].ProcessVariables.Parameter_DINT [#].ID	Process variables Parameter ID	DINT		X
Status	Recipe [#].ProcessVariables.Parameter_DINT [#].Name	Process variables Name of parameter	STRING		X
Status	Recipe [#].ProcessVariables.Parameter_DINT [#].Unit	Process variables Unit of measure	STRING[5]		X
Status	Recipe [#].ProcessVariables.Parameter_DINT [#].Value	Process variables Value of parameter	DINT		X
Status	Recipe [#].Ingredients ²⁸ .	Ingredients variables			X
Status	Recipe [#].Ingredients.Parameter_REAL [#]	Ingredients/parameter	Array Structure		X
Status	Recipe [#].Ingredients.Parameter_REAL [#].ID	Ingredients Parameter ID	DINT		X
Status	Recipe [#].Ingredients.Parameter_REAL [#].Name	Ingredients Name of parameter	STRING		X
Status	Recipe [#].Ingredients.Parameter_REAL [#].Unit	Ingredients Unit of measure	STRING[5]		X
Status	Recipe [#].Ingredients.Parameter_REAL [#].Value	Ingredients Value of parameter	REAL		X
Status	Recipe [#].Ingredients.Parameter_STRING [#]	Ingredients Machine data/parameter	Array Structure		X
Status	Recipe [#].Ingredients.Parameter_STRING [#].ID	Ingredients Parameter ID	DINT		X
Status	Recipe [#].Ingredients.Parameter_STRING [#].Name	Ingredients Name of parameter	STRING		X
Status	Recipe [#].Ingredients.Parameter_STRING [#].Unit	Ingredients Unit of measure	STRING[5]		X
Status	Recipe [#].Ingredients.Parameter_STRING [#].Value	Ingredients Value of parameter	STRING		X
Status	Recipe [#].Ingredients.Parameter_LREAL [#]	Ingredients Machine data/parameter	Array Structure		X
Status	Recipe [#].Ingredients.Parameter_LREAL [#].ID	Ingredients Parameter ID	DINT		X
Status	Recipe [#].Ingredients.Parameter_LREAL [#].Name	Ingredients Name of parameter	STRING		X
Status	Recipe [#].Ingredients.Parameter_LREAL [#].Unit	Ingredients Unit of measure	STRING[5]		X
Status	Recipe [#].Ingredients.Parameter_LREAL [#].Value	Ingredients Value of parameter	LREAL		X
Status	Recipe [#].Ingredients.Parameter_DINT [#]	Ingredients Machine data/parameter	Array Structure		X
Status	Recipe [#].Ingredients.Parameter_DINT [#].ID	Ingredients Parameter ID	DINT		X
Status	Recipe [#].Ingredients.Parameter_DINT [#].Name	Ingredients Name of parameter	STRING		X
Status	Recipe [#].Ingredients.Parameter_DINT [#].Unit	Ingredients Unit of measure	STRING[5]		X
Status	Recipe [#].Ingredients.Parameter_DINT [#].Value	Ingredients Value of parameter	DINT		X

X = Supported.

²⁶Status.Recipe 呈现的数据可以用于 Centerlining 功能性。

²⁷ProcessVariables 结构可以用于 Unit/Machine 处理特定配方所需的特定设置点。ProcessVariables 可能包括速度、压力、真空、时间设置点、限制等。Unit/Machine 应使用 ProcessVariables 对按配方生产的产品进行配置。

²⁸成分结构可以用于保存 Unit/Machine 根据配方加工特定产品时使用的原材料信息。成分可能包含原材料数量。

表 20：最小 ADMIN PackTag²⁹

PackTag type	PackTag	Example of End user term	Datatype IEC 61131-3	TR 88.00.02 Minimum tags	End user Minimum tags
Admin	Alarm[#]	Alarm	Array Structure		X
Admin	Alarm [#].Trigger	Trigger	BOOL		X
Admin	Alarm [#].ID	ID	DINT		X
Admin	Alarm [#].Value	值	DINT		X
Admin	Warning[#]	Warning	Array Structure		X
Admin	Warning[#].Trigger	Trigger	BOOL		X
Admin	Warning[#].ID	ID	DINT		X
Admin	Warning[#].Value	值	DINT		X
Admin	ProductData [#]	Product data array	Array Structure		X
Admin	ProductData [#].ID	Product identification	DINT		X
Admin	ProductData [#].ProcessedCount	Total count processed products	DINT	X	X
Admin	ProductData [#].DefectiveCount	Bad counted products	DINT	X	X
	StopReason	StopReasons variables	Array Structure		X
Admin	StopReason.ID	Event and stop reason	DINT	X	X
Admin	StopReason.Value	Detailed Error Information	DINT		X

X = Supported.

表 21：最小 COMMAND PackTag³⁰

PackTag type	PackTag	Example of End user term	Datatype IEC 61131-3	TR 88.00.02 Minimum tags	End user Minimum tags
Command	UnitMode	Mode	DINT	X	X
Command	UnitModeChangeRequest	Change mode	BOOL	X	X
Command	MachSpeed ³¹	Mach Speed	REAL	X	X
Command	CntrlCmd	Command	DINT	X	X
Command	CmdChangeRequest	Change command	BOOL	X	X
Command	Parameter_REAL [#]	Machine data/parameter	Array Structure		X
Command	Parameter_REAL [#].ID	Parameter ID	DINT		X
Command	Parameter_REAL [#].Name	Name of parameter	STRING		X
Command	Parameter_REAL [#].Unit	Unit of measure	STRING[5]		X
Command	Parameter_REAL [#].Value	Value of parameter	REAL		X
Command	Parameter_STRING [#]	Machine data/parameter	Array Structure		X
Command	Parameter_STRING [#].ID	Parameter ID	DINT		X
Command	Parameter_STRING [#].Name	Name of parameter	STRING		X
Command	Parameter_STRING [#].Unit	Unit of measure	STRING[5]		X
Command	Parameter_STRING [#].Value	Value of parameter	STRING		X
Command	Parameter_LREAL [#]	Machine data/parameter	Array Structure		X
Command	Parameter_LREAL [#].ID	Parameter ID	DINT		X
Command	Parameter_LREAL [#].Name	Name of parameter	STRING		X
Command	Parameter_LREAL [#].Unit	Unit of measure	STRING[5]		X
Command	Parameter_LREAL [#].Value	Value of parameter	LREAL		X
Command	Parameter_DINT [#]	Machine data/parameter	Array Structure		X
Command	Parameter_DINT [#].ID	Parameter ID	DINT		X
Command	Parameter_DINT [#].Name	Name of parameter	STRING		X
Command	Parameter_DINT [#].Unit	Unit of measure	STRING[5]		X
Command	Parameter_DINT [#].Value	Value of parameter	DINT		X
Command	SelectedRecipe	Number of Recipe	DINT		X
Command	RecipeChangeRequest	Requested Recipe No	DINT		X

29 ANSI/ISA-TR88.00.02-2022 是所有 PackTag 的参考。

30 ANSI/ISA-TR88.00.02-2022 是所有 PackTag 的参考。

31 如果 Unit/Machine 可以在同一条生产线上运行具有不同内包装尺寸的不同产品，MachSpeed 需要作为机器配方参数的一部分来计算及查看，以帮助操作员了解机器及其速度。原因是，以小型进给系统为例，它的速度可能会随着内包装尺寸的变化而变化。

PackTag type	PackTag	Example of End user term	Datatype IEC 61131-3	TR 88.00.02 Minimum tags	End user Minimum tags
Command	Recipe ³² [#]	Recipe structure	Array Structure		X
Command	Recipe [#].ID	Recipe ID	DINT		X
Command	Recipe [#].Name	Recipe Name	STRING		X
Command	Recipe [#].Unit	Unit of Measure (E.g., Pc, Crtn, Case, ...)	STRING[5]		X
Command	Recipe [#].PrimaryQty	Primary	REAL		X
Command	Recipe [#].ProcessVariables ³³ .	Process variables			X
Command	Recipe [#].ProcessVariables.Parameter_REAL [#]	Process variables Machine data/parameter	Array Structure		X
Command	Recipe [#].ProcessVariables.Parameter_REAL [#].ID	Process variables Parameter ID	DINT		X
Command	Recipe [#].ProcessVariables.Parameter_REAL [#].Name	Process variables Name of parameter	STRING		X
Command	Recipe [#].ProcessVariables.Parameter_REAL [#].Unit	Process variables Unit of measure	STRING[5]		X
Command	Recipe [#].ProcessVariables.Parameter_REAL [#].Value	Process variables Value of parameter	REAL		X
Command	Recipe [#].ProcessVariables.Parameter_STRING [#]	Process variables Machine data/parameter	Array Structure		X
Command	Recipe [#].ProcessVariables.Parameter_STRING [#].ID	Process variables Parameter ID	DINT		X
Command	Recipe [#].ProcessVariables.Parameter_STRING [#].Name	Process variables Name of parameter	STRING		X
Command	Recipe [#].ProcessVariables.Parameter_STRING [#].Unit	Process variables Unit of measure	STRING[5]		X
Command	Recipe [#].ProcessVariables.Parameter_STRING [#].Value	Process variables Value of parameter	STRING		X
Command	Recipe [#].ProcessVariables.Parameter_LREAL [#]	Process variables Machine data/parameter	Array Structure		X
Command	Recipe [#].ProcessVariables.Parameter_LREAL [#].ID	Process variables Parameter ID	DINT		X
Command	Recipe [#].ProcessVariables.Parameter_LREAL [#].Name	Process variables Name of parameter	STRING		X
Command	Recipe [#].ProcessVariables.Parameter_LREAL [#].Unit	Process variables Unit of measure	STRING[5]		X
Command	Recipe [#].ProcessVariables.Parameter_LREAL [#].Value	Process variables Value of parameter	LREAL		X
Command	Recipe [#].ProcessVariables.Parameter_DINT [#]	Process variables Machine data/parameter	Array Structure		X
Command	Recipe [#].ProcessVariables.Parameter_DINT [#].ID	Process variables Parameter ID	DINT		X
Command	Recipe [#].ProcessVariables.Parameter_DINT [#].Name	Process variables Name of parameter	STRING		X
Command	Recipe [#].ProcessVariables.Parameter_DINT [#].Unit	Process variables Unit of measure	STRING[5]		X
Command	Recipe [#].ProcessVariables.Parameter_DINT [#].Value	Process variables Value of parameter	DINT		X
Command	Recipe [#].Ingredients ³⁴	Ingredients variables			X
Command	Recipe [#].Ingredients.Parameter_REAL [#]	Ingredients/parameter	Array Structure		X
Command	Recipe [#].Ingredients.Parameter_REAL [#].ID	Ingredients Parameter ID	DINT		X
Command	Recipe [#].Ingredients.Parameter_REAL [#].Name	Ingredients Name of parameter	STRING		X
Command	Recipe [#].Ingredients.Parameter_REAL [#].Unit	Ingredients Unit of measure	STRING[5]		X
Command	Recipe [#].Ingredients.Parameter_REAL [#].Value	Ingredients Value of parameter	REAL		X
Command	Recipe [#].Ingredients.Parameter_STRING [#]	Ingredients Machine data/parameter	Array Structure		X
Command	Recipe [#].Ingredients.Parameter_STRING [#].ID	Ingredients Parameter ID	DINT		X
Command	Recipe [#].Ingredients.Parameter_STRING [#].Name	Ingredients Name of parameter	STRING		X
Command	Recipe [#].Ingredients.Parameter_STRING [#].Unit	Ingredients Unit of measure	STRING[5]		X
Command	Recipe [#].Ingredients.Parameter_STRING [#].Value	Ingredients Value of parameter	STRING		X
Command	Recipe [#].Ingredients.Parameter_LREAL [#]	Ingredients Machine data/parameter	Array Structure		X
Command	Recipe [#].Ingredients.Parameter_LREAL [#].ID	Ingredients Parameter ID	DINT		X
Command	Recipe [#].Ingredients.Parameter_LREAL [#].Name	Ingredients Name of parameter	STRING		X
Command	Recipe [#].Ingredients.Parameter_LREAL [#].Unit	Ingredients Unit of measure	STRING[5]		X
Command	Recipe [#].Ingredients.Parameter_LREAL [#].Value	Ingredients Value of parameter	LREAL		X
Command	Recipe [#].Ingredients.Parameter_DINT [#]	Ingredients Machine data/parameter	Array Structure		X
Command	Recipe [#].Ingredients.Parameter_DINT [#].ID	Ingredients Parameter ID	DINT		X
Command	Recipe [#].Ingredients.Parameter_DINT [#].Name	Ingredients Name of parameter	STRING		X
Command	Recipe [#].Ingredients.Parameter_DINT [#].Unit	Ingredients Unit of measure	STRING[5]		X
Command	Recipe [#].Ingredients.Parameter_DINT [#].Value	Ingredients Value of parameter	DINT		X

X = Supported.

³²Command.Recipe 用于配置 Unit/Machine，并可用于监控控制系统的订单处理功能。

³³ProcessVariables 结构可以用于 Unit/Machine 处理特定配方所需的特定设置点。ProcessVariables 可能包括速度、压力、真空、时间设置点、限制等。Unit/Machine 应使用 ProcessVariables 对按配方生产的产品进行配置。

³⁴成分结构可以用于保存 Unit/Machine 根据配方加工特定产品时使用的原材料信息。成分可能包含原材料数量。

11.1 PACKTAG 说明和定义

有关 PackTag 的详细信息, 请参见 *ANSI/ISA-TR88.00.02-2022, Machine and Unit/Machine States, ANSI/ISA-88.00.01* 实施示例。

11.2 PACKTAG 2015 到 PACKTAG 2022 的变迁

以下是 PackTag 2015 和 PackTag 2022 之间差异的概述。

下表显示了符合 ISA-TR88.00.02-2015 标准的 Status、Admin 和 Command 标签。单个表格比较了 PackTag 2015 和 PackTag 2022。表格中的列指定了以下文字:

- **ISA-TR88-2015 PackTags:** 列中 PackTag 是 PackTag 2015。

- **ISA-TR88-2022 PackTags alignment**
 - 带有 x 和绿色背景的单元格表示 ISA-TR88.00.02-2022 符合 ISA-TR88.00.02-2015。
 - 带有 (-) 和红色背景的单元格表示其不包含在 ISA-TR88.00.02-2022 中。
 - 带有 (x) 和黄色背景的单元格表示内容包含在 ISA-TR88.00.02-2022 中，但结构或数据类型不同。
- **Comments:** 描述两个标准 PackTag 之间的差异。

表 22: Status PackTag

ISA-TR88-2015 PackTags	ISA-TR88-2022 PackTags alignment	Comments
UnitName	(x)	更改为 PACKML v2022
UnitName.Status	x	
UnitName.Status.UnitModeCurrent	x	
UnitName.Status.UnitModeRequested	x	
UnitName.Status.UnitModeChangeInProcess	x	
UnitName.Status.StateCurrent	x	
UnitName.Status.StateRequested	x	
UnitName.Status.StateChangeInProcess	x	
UnitName.Status.MachSpeed	x	
UnitName.Status.CurMachSpeed	x	
UnitName.Status.MaterialInterlock[#]	x	
UnitName.Status.EquipmentInterlock	x	
UnitName.Status.EquipmentInterlock.Blocked	x	
UnitName.Status.EquipmentInterlock.Starved	x	
UnitName.Status.RemoteInterface[#]	-	ISA-TR88.00.02-2022 中不包括远程接口。
UnitName.Status.RemoteInterface[#.Number]	-	
UnitName.Status.RemoteInterface[#.ControlCmdNumber]	-	
UnitName.Status.RemoteInterface[#.CmdValue]	-	
UnitName.Status.RemoteInterface[#.Parameter[#]]	-	
UnitName.Status.RemoteInterface[#.Parameter[#].ID]	-	
UnitName.Status.RemoteInterface[#.Parameter[#].Name]	-	
UnitName.Status.RemoteInterface[#.Parameter[#].Unit]	-	
UnitName.Status.RemoteInterface[#.Parameter[#].Value]	-	
UnitName.Status.Parameter[#]	(x)	参数的 PackTag 阵列被分成需要包含的不同类型的变量。这些类型是：
UnitName.Status.Parameter[#].ID	(x)	UnitName.Status.Parameter_REAL[#] UnitName.Status.Parameter_STRING[#]
UnitName.Status.Parameter[#].Name	(x)	UnitName.Status.Parameter_LREAL[#] UnitName.Status.Parameter_DINT[#]
UnitName.Status.Parameter[#].Unit	(x)	
UnitName.Status.Parameter[#].Value	(x)	
UnitName.Status.Product[#]	(x)	PackTag 阵列 “Product” 被 UnitName.Status.Recipe[#] 替代
UnitName.Status.Product[#.ProductID]	(x)	更改为 UnitName.Status.Recipe[#].ID
UnitName.Status.Product[#].ProcessVariables[#]	(x)	PackTag 阵列变更为
UnitName.Status.Product[#].ProcessVariables[#].ID	(x)	UnitName.Status.Recipe[#].ProcessVariables.Parameter_REAL[#], UnitName.Status.Recipe[#].ProcessVariables.Parameter_STRING[#], UnitName.Status.Recipe[#].ProcessVariables.Parameter_LREAL[#]
UnitName.Status.Product[#].ProcessVariables[#].Name	(x)	UnitName.Status.Recipe[#].ProcessVariables.Parameter_DINT[#]
UnitName.Status.Product[#].ProcessVariables[#].Unit	(x)	
UnitName.Status.Product[#].ProcessVariables[#].Value	(x)	
UnitName.Status.Product[#].Ingredients[#]	(x)	PackTag 阵列变更为
UnitName.Status.Product[#].Ingredients[#].IngredientID	(x)	UnitName.Status.Recipe[#].Ingredients.Parameter_REAL[#], UnitName.Status.Recipe[#].Ingredients.Parameter_STRING[#], UnitName.Status.Recipe[#].Ingredients.Parameter_LREAL[#]
UnitName.Status.Product[#].Ingredients[#].Parameter[#]	(x)	UnitName.Status.Recipe[#].Ingredients.Parameter_DINT[#]
UnitName.Status.Product[#].Ingredients[#].Parameter[#].ID	(x)	
UnitName.Status.Product[#].Ingredients[#].Parameter[#].Name	(x)	
UnitName.Status.Product[#].Ingredients[#].Parameter[#].Unit	(x)	
UnitName.Status.Product[#].Ingredients[#].Parameter[#].Value	(x)	

表 23: Admin PackTag

ISA-TR88-2015 PackTags	ISA-TR88-2022 PackTags alignment	Comments
UnitName	(x)	更改为 PACKML v2022
UnitName.Admin	x	
UnitName.Admin.Parameter[#]	(x)	参数的 PackTag 阵列被分成需要包含的不同类型的变量。这些类型是:
UnitName.Admin.Parameter[#].ID	(x)	UnitName.Admin.Parameter_REAL[#]
UnitName.Admin.Parameter[#].Name	(x)	UnitName.Admin.Parameter_STRING[#]
UnitName.Admin.Parameter[#].Unit	(x)	UnitName.Admin.Parameter_LREAL[#]
UnitName.Admin.Parameter[#].Value	(x)	UnitName.Admin.Parameter_DINT[#]
UnitName.Admin.Alarm[#]	x	
UnitName.Admin.Alarm[#].Trigger	x	
UnitName.Admin.Alarm[#].ID	x	
UnitName.Admin.Alarm[#].Value	x	
UnitName.Admin.Alarm[#].Message	x	
UnitName.Admin.Alarm[#].Category	x	
UnitName.Admin.Alarm[#].DateTime[#]	(x)	DateTime 和 AckDateTime 不是阵列, 而变为结构体 <i>UnitName.Admin.Alarm[#].DateTime.Year</i> <i>UnitName.Admin.Alarm[#].DateTime.Month</i> <i>UnitName.Admin.Alarm[#].DateTime.Day</i> <i>UnitName.Admin.Alarm[#].DateTime.Hour</i> <i>UnitName.Admin.Alarm[#].DateTime.Minute</i> <i>UnitName.Admin.Alarm[#].DateTime.Second</i>
UnitName.Admin.AlarmExtent	x	
UnitName.Admin.AlarmHistory[#]	x	
UnitName.Admin.AlarmHistory[#].Trigger	x	
UnitName.Admin.AlarmHistory[#].ID	x	
UnitName.Admin.AlarmHistory[#].Value	x	
UnitName.Admin.AlarmHistory[#].Message	x	
UnitName.Admin.AlarmHistory[#].Category	x	
UnitName.Admin.AlarmHistory[#].DateTime[#]	(x)	DateTime 和 AckDateTime 不是阵列, 而变为结构体 <i>UnitName.Admin.AlarmHistory[#].DateTime.Year</i> <i>UnitName.Admin.AlarmHistory[#].DateTime.Month</i> <i>UnitName.Admin.AlarmHistory[#].DateTime.Day</i> <i>UnitName.Admin.AlarmHistory[#].DateTime.Hour</i> <i>UnitName.Admin.AlarmHistory[#].DateTime.Minute</i> <i>UnitName.Admin.AlarmHistory[#].DateTime.Second</i>
UnitName.Admin.StopReason	-	StopReason Array 变更成一个简单的数据类型, 无阵列。 <i>UnitName.Admin.StopReason</i>
UnitName.Admin.StopReason[#]	(x)	
UnitName.Admin.StopReason[#].Trigger	(x)	
UnitName.Admin.StopReason[#].ID	(x)	
UnitName.Admin.StopReason[#].Value	(x)	DateTime 和 AckDateTime 不是阵列, 而变为结构体 <i>UnitName.Admin.StopReason.DateTime.Year</i> <i>UnitName.Admin.StopReason.DateTime.Month</i> <i>UnitName.Admin.StopReason.DateTime.Day</i> <i>UnitName.Admin.StopReason.DateTime.Hour</i> <i>UnitName.Admin.StopReason.DateTime.Minute</i> <i>UnitName.Admin.StopReason.DateTime.Second</i>
UnitName.Admin.StopReason[#].Message	(x)	
UnitName.Admin.StopReason[#].Category	(x)	
UnitName.Admin.StopReason[#].DateTime[#]	(x)	
UnitName.Admin.StopReason[#].AckDateTime[#]	(x)	
UnitName.Admin.StopReasonExtent	-	Removed - 不再是 PackTag 的一部分
UnitName.Admin.Warning[#]	x	
UnitName.Admin.Warning[#].Trigger	x	
UnitName.Admin.Warning[#].ID	x	
UnitName.Admin.Warning[#].Value	x	
UnitName.Admin.Warning[#].Message	x	
UnitName.Admin.Warning[#].Category	x	
UnitName.Admin.Warning[#].DateTime[#]	(x)	DateTime 和 AckDateTime 不是阵列, 而变为结构体 <i>UnitName.Admin.Warning.DateTime.Year</i> <i>UnitName.Admin.Warning.DateTime.Month</i> <i>UnitName.Admin.Warning.DateTime.Day</i> <i>UnitName.Admin.Warning.DateTime.Hour</i> <i>UnitName.Admin.Warning.DateTime.Minute</i> <i>UnitName.Admin.Warning.DateTime.Second</i>
UnitName.Admin.Warning[#].AckDateTime[#]	(x)	
UnitName.Admin.WarningExtent	x	

ISA-TR88-2015 PackTags	ISA-TR88-2022 PackTags alignment	Comments
UnitName.Admin.ModeCurrentTime[#]	(x)	StopReason Array 变更成一个简单的数据类型，无阵列。 UnitName.Admin.StopReason
UnitName.Admin.ModeCumulativeTime[#]	(x)	
UnitName.Admin.StateCurrentTime[#,#]	(x)	
UnitName.Admin.StateCumulativeTime[#,#]	(x)	
UnitName.Admin.ProdConsumedCount[#]	(x)	结构体变更为 <i>UnitName.Admin.ProductData[#].</i>
UnitName.Admin.ProdConsumedCount[#,].ID	(x)	
UnitName.Admin.ProdConsumedCount[#,].Name	(x)	
UnitName.Admin.ProdConsumedCount[#,].Unit	(x)	计数器变更为 <i>UnitName.Admin.ProductData[#].PrimaryQty</i>
UnitName.Admin.ProdConsumedCount[#,].Count	(x)	<i>UnitName.Admin.ProductData[#].ConsumedCount</i>
UnitName.Admin.ProdProcessedCount[#]	(x)	<i>UnitName.Admin.ProductData[#].ProcessedCount</i>
(x)	(x)	<i>UnitName.Admin.ProductData[#].DefectiveCount</i>
UnitName.Admin.ProdProcessedCount[#,].Name	(x)	<i>UnitName.Admin.ProductData[#].AccConsumedCount</i>
UnitName.Admin.ProdProcessedCount[#,].Unit	(x)	<i>UnitName.Admin.ProductData[#].AccProcessedCount</i>
UnitName.Admin.ProdProcessedCount[#,].Count	(x)	<i>UnitName.Admin.ProductData[#].AccDefectiveCount</i>
UnitName.Admin.ProdProcessedCount[#,].AccCount	(x)	
UnitName.Admin.ProdDefectiveCount[#]	(x)	
UnitName.Admin.ProdDefectiveCount[#,].ID	(x)	
UnitName.Admin.ProdDefectiveCount[#,].Name	(x)	
UnitName.Admin.ProdDefectiveCount[#,].Unit	(x)	
UnitName.Admin.ProdDefectiveCount[#,].Count	(x)	
UnitName.Admin.ProdDefectiveCount[#,].AccCount	(x)	
UnitName.Admin.ProdDefectiveCount[0].Count	(x)	
UnitName.Admin.ProdDefectiveCount[1].Count	(x)	
UnitName.Admin.ProdDefectiveCount[2].Count	(x)	
UnitName.Admin.ProdDefectiveCount[3].Count	(x)	
UnitName.Admin.ProdDefectiveCount[4].Count	(x)	
UnitName.Admin.ProdDefectiveCount[5].Count	(x)	
UnitName.Admin.ProdDefectiveCount[6].Count	(x)	
UnitName.Admin.AccTimeSinceReset	(x)	
UnitName.Admin.MachDesignSpeed	X	
UnitName.Admin.StatesDisabled	(x)	变更为两个参数： <i>UnitName.Admin.DisabledStatesCfg[#]</i> <i>UnitName.Admin.CurDisabledStates</i> 同时增加模式配置： <i>UnitName.Admin.ModeTransitionCfg[#]</i> <i>UnitName.Admin.EnabledModesCfg</i>
UnitName.Admin.PLCDDateTime[#]	(x)	PLCDDateTime 不是阵列，而变为结构体 <i>UnitName.Admin.Warning.DateTime.Year</i> <i>UnitName.Admin.Warning.DateTime.Month</i> <i>UnitName.Admin.Warning.DateTime.Day</i> <i>UnitName.Admin.Warning.DateTime.Hour</i> <i>UnitName.Admin.Warning.DateTime.Minute</i> <i>UnitName.Admin.Warning.DateTime.Second</i>

表 24: Command PackTags

ISA-TR88-2015 PackTags	ISA-TR88-2022 PackTags alignment	Comments
UnitName	(x)	更改为 PACKML v2022
UnitName.Command	x	
UnitName.Command.UnitMode	x	
UnitName.Command.UnitModeChangeRequest	x	
UnitName.Command.MachSpeed	x	
UnitName.Command.MaterialInterlock[#]	x	
UnitName.Command.CntrlCmd	x	
UnitName.Command.CmdChangeRequest	x	
UnitName.Command.RemoteInterface[#]	-	从 ISA-TR88.00.02-2022 中删除了远程接口。
UnitName.Command.RemoteInterface[#].Number	-	
UnitName.Command.RemoteInterface[#].ControlCmdNumber	-	
UnitName.Command.RemoteInterface[#].CmdValue	-	
UnitName.Command.RemoteInterface[#].Parameter[#]	-	
UnitName.Command.RemoteInterface[#].Parameter[#].ID	-	
UnitName.Command.RemoteInterface[#].Parameter[#].Name	-	
UnitName.Command.RemoteInterface[#].Parameter[#].Unit	-	
UnitName.Command.RemoteInterface[#].Parameter[#].Value	-	
UnitName.Command.Parameter[#]	(x)	参数的 PackTag 阵列被分成需要包含的不同类型的变量。 这些类型是:
UnitName.Command.Parameter[#].ID	(x)	UnitName.Command.Parameter_REAL[#] UnitName.Command.Parameter_STRING[#] UnitName.Command.Parameter_LREAL[#] UnitName.Command.Parameter_DINT[#]
UnitName.Command.Parameter[#].Name	(x)	
UnitName.Command.Parameter[#].Unit	(x)	
UnitName.Command.Parameter[#].Value	(x)	
UnitName.Command.Product[#]	(x)	PackTag 阵列 “Product” 被 UnitName.Command.Recipe[#] 替代
UnitName.Command.Product[#].ProductId	(x)	更改为 UnitName.Command.Recipe[#].ID
UnitName.Command.Product[#].ProcessVariables[#]	(x)	PackTag 阵列变更为 UnitName.Command.Recipe[#].ProcessVariables.Parameter_REAL[#], UnitName.Command.Recipe[#].ProcessVariables.Parameter_STRING[#], UnitName.Command.Recipe[#].ProcessVariables.Parameter_LREAL[#] UnitName.Command.Recipe[#].ProcessVariables.Parameter_DINT[#]
UnitName.Command.Product[#].ProcessVariables[#].ID	(x)	
UnitName.Command.Product[#].ProcessVariables[#].Name	(x)	
UnitName.Command.Product[#].ProcessVariables[#].Unit	(x)	
UnitName.Command.Product[#].ProcessVariables[#].Value	(x)	
UnitName.Command.Product[#].Ingredients[#]	(x)	PackTag 阵列变更为 UnitName.Command.Recipe[#].Ingredients.Parameter_REAL[#], UnitName.Command.Recipe[#].Ingredients.Parameter_STRING[#], UnitName.Command.Recipe[#].Ingredients.Parameter_LREAL[#] UnitName.Command.Recipe[#].Ingredients.Parameter_DINT[#]
UnitName.Command.Product[#].Ingredients[#].IngredientID	(x)	
UnitName.Command.Product[#].Ingredients[#].Parameter[#]	(x)	
UnitName.Command.Product[#].Ingredients[#].Parameter[#].ID	(x)	
UnitName.Command.Product[#].Ingredients[#].Parameter[#].Name	(x)	
UnitName.Command.Product[#].Ingredients[#].Parameter[#].Unit	(x)	
UnitName.Command.Product[#].Ingredients[#].Parameter[#].Value	(x)	

12. UNIT CONTROLLER FUNCTIONALITY 示例

本节的要素中说明了 Unit Controller functionality 的八个主要功能:

1. Prepare Unit/Machine (Prepare for production)
2. Start Unit/Machine (Start production)
3. Stop Unit/Machine (Stop production)
 - a. Stop Unit/Machine (Stop production)
 - b. Complete Unit/Machine (Complete production)
4. Pause on Unit/Machine
5. Stop and restart Unit/Machine
6. Download new machine parameters (Order change)
7. Error and warnings on Unit/Machine
8. Safety circuit activated

下图以一台 Unit 为例, 说明了从上游 Unit 到下游 Unit 的物料流。

12.1 SYNTAX 和 SYMBOLIC 说明

下图说明了在相关用例场景中使用的 syntax 和 symbolic。

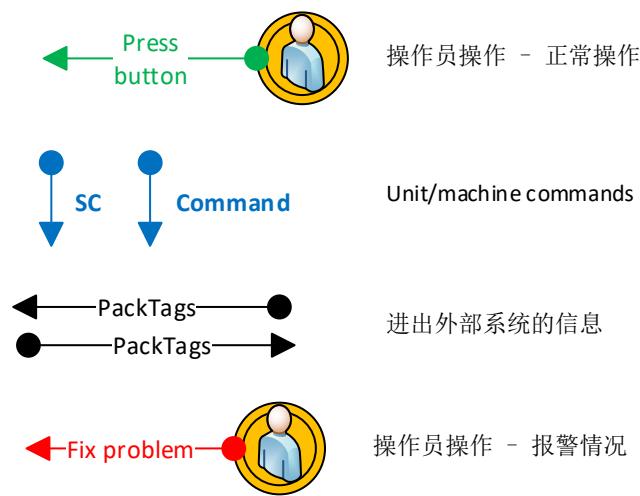


图 36: 用例场景中的 Syntax 和 Symbolic

12.2 准备 UNIT/MACHINE

操作员指示要做好生产准备的 Unit，并将该 Unit 经由 Resetting 状态切换到 IDLE 状态。操作员需要确认所有安全条件都已满足而且该 Unit 做好了生产准备。

Prepare unit/machine

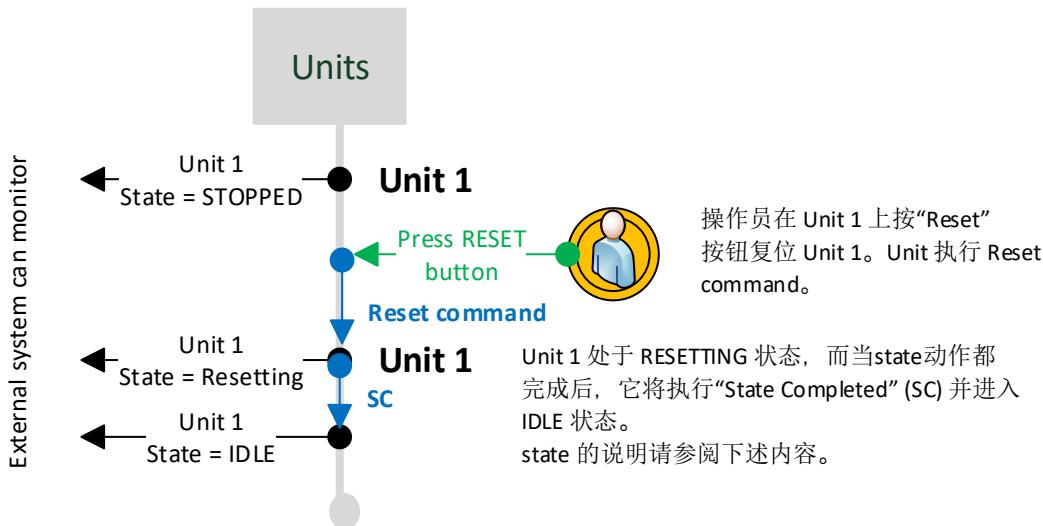


图 37：准备生产用 Unit/Machine

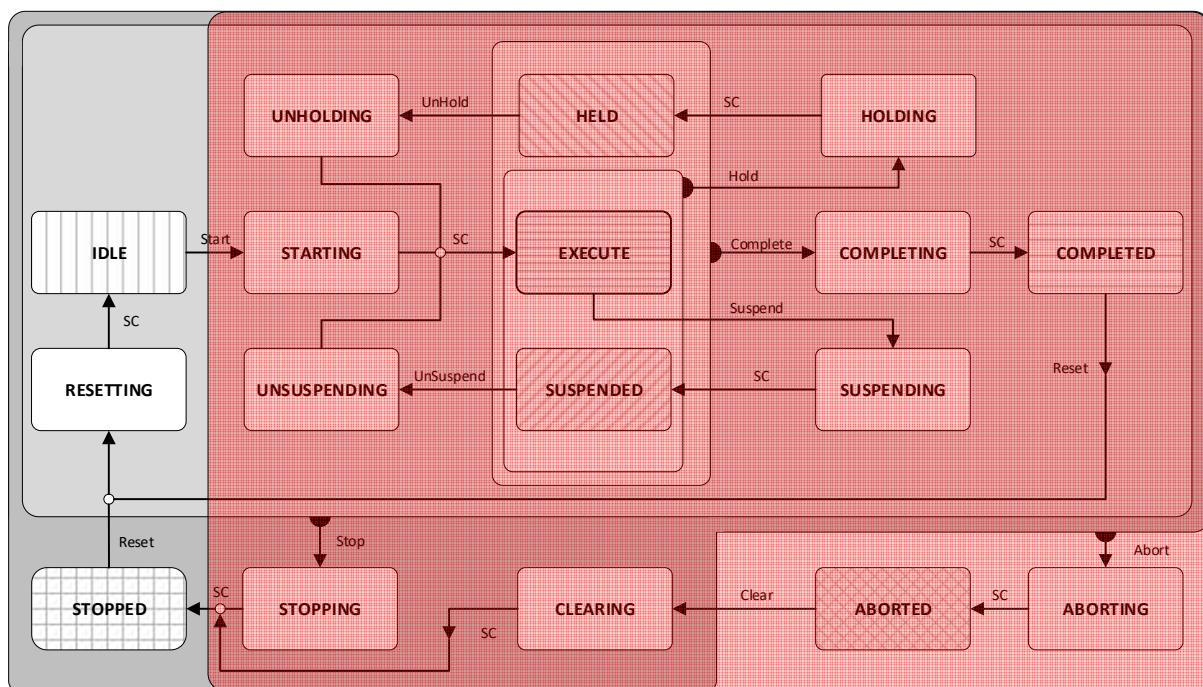


图 38：PackML state model 中的状态聚焦于 Prepare Unit/Machine

12.3 启动 UNIT/MACHINE

操作员在 Unit Panel/HMI 上或外部系统启动 Unit/Machine 进行生产。操作员在 Unit/Machine 上启动所选的程序后，相关参数会加载到 Unit 中。Unit/Machine 参数可在 Unit/Machine Panel/HMI 上输入，也可以从外部系统下载到 Unit/Machine。

操作员或外部系统根据 PackML 标准启动 Unit/Machine。

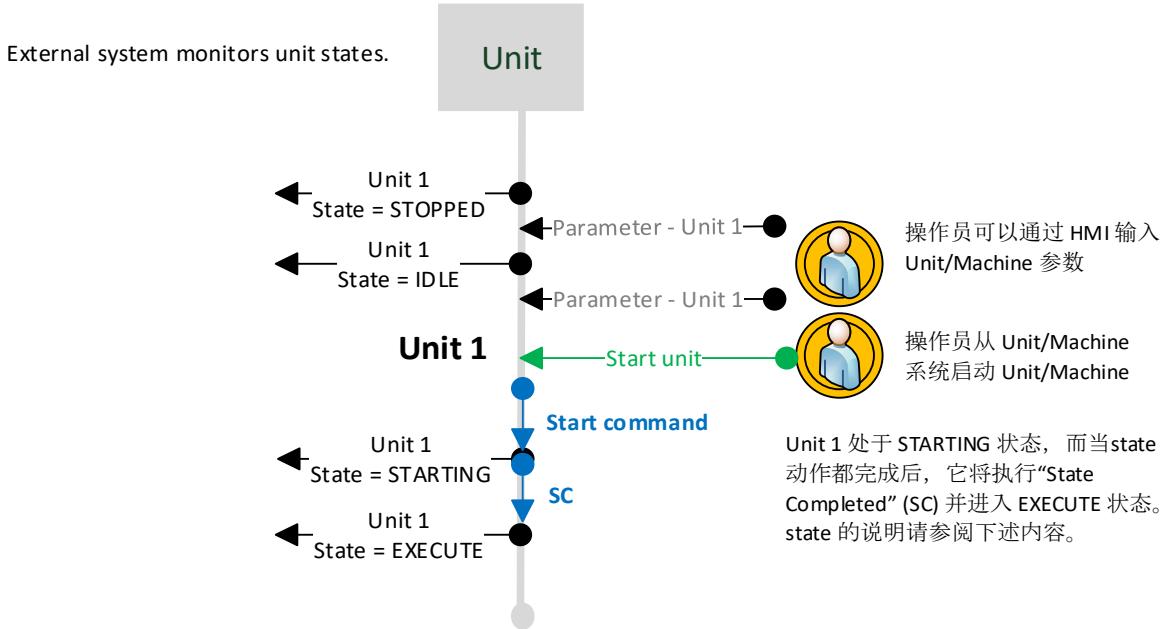


图 39：从 Unit Panel/HMI 启动 Unit/Machine

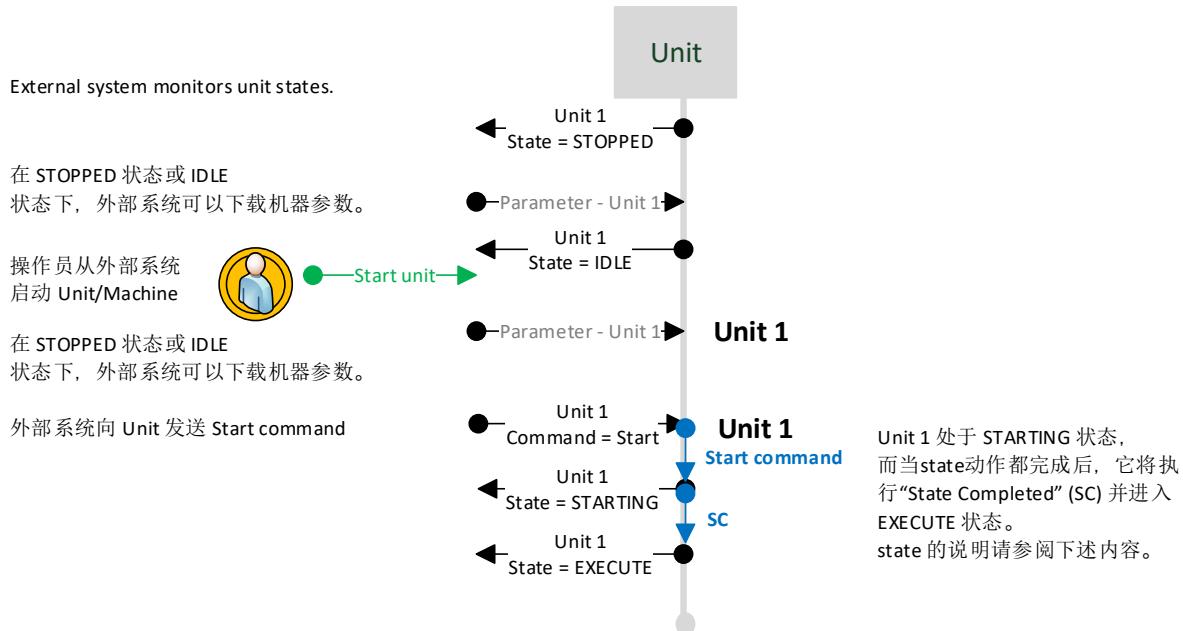


图 40：从外部系统启动 Unit/Machine

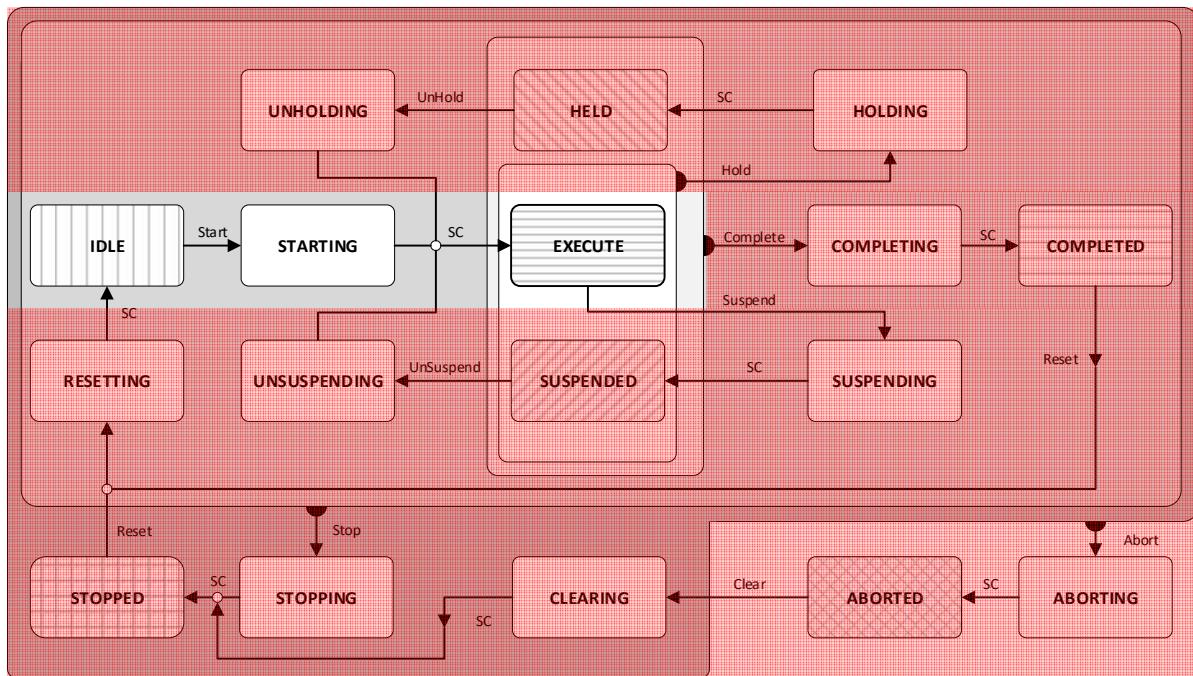


图 41：PackML state model 中的状态聚焦于启动 Unit/Machine

12.4 从 UNIT/MACHINE PANEL/HMI 停止 UNIT/MACHINE

产品计数器达到机器参数中指定值后，Unit 可以切换到 COMPLETE 状态。Unit 将完成处理、清空并停止产品生产，然后进入 COMPLETE 状态。

提示：Unit 完成生产时，如果进口/进料区仍有产品，则有必要定义 Unit 上发生了什么。

提示：Unit / Machine 的机器供应商和最终用户需要定义在完成生产订单（产品计数器达到了生产订单中指定的数量）后 Unit / Machine 是否需要停止。这可能取决于 Unit / Machine 参数何时由操作员下载至或手动输入至 Unit / Machine。

提示：Unit / Machine 因为缺料处于 Suspended 状态时，它只能在操作员（在 Unit / Machine 上或通过监控控制系统）发出 Stop 指令后完成生产订单。

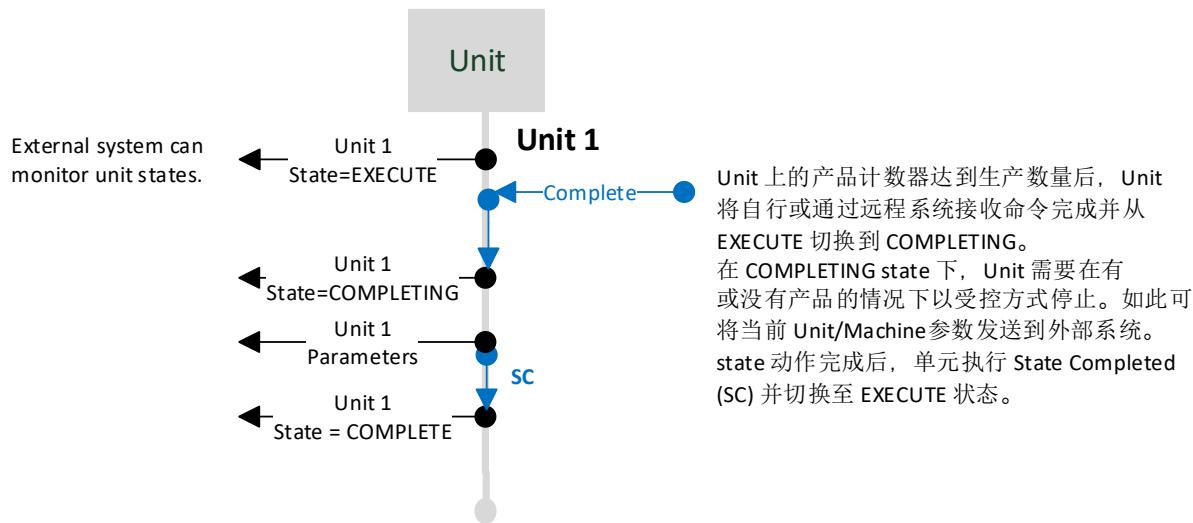


图 42：从 Unit 完成

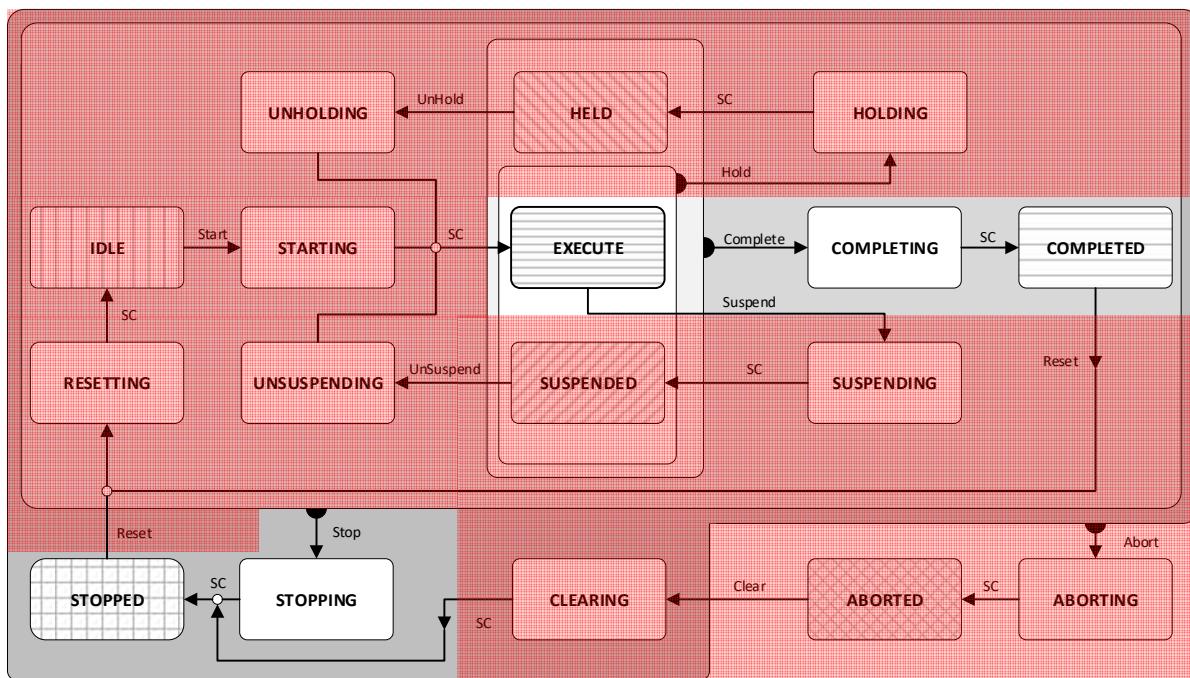


图 43: PackML state model 中状态聚焦于从 Unit 或远程控制完成

12.5 从外部系统停止 UNIT/MACHINE

外部系统可从 Unit 持续获取已生产产品的数量信息。外部系统也可掌握正在生产的产品数量。外部系统可以将已生产产品的数量与从生产订单数据中收到的参数值加以比较。当已生产的产品数量等于该参数值 (100%) 时，外部系统可以向操作员发送一条消息。

操作员随后可以从外部系统停止 Unit 或停止生产。Unit 随后停止，进入 STOPPING 状态，清空（如有必要）和停止生产并进入 STOPPED 状态。Unit 停止并完成生产后，Unit/Machine 参数可以上传到外部系统。

提示: Unit 停止时，如果进口/进料区仍有产品，则有必要确认 Unit 上发生了什么。

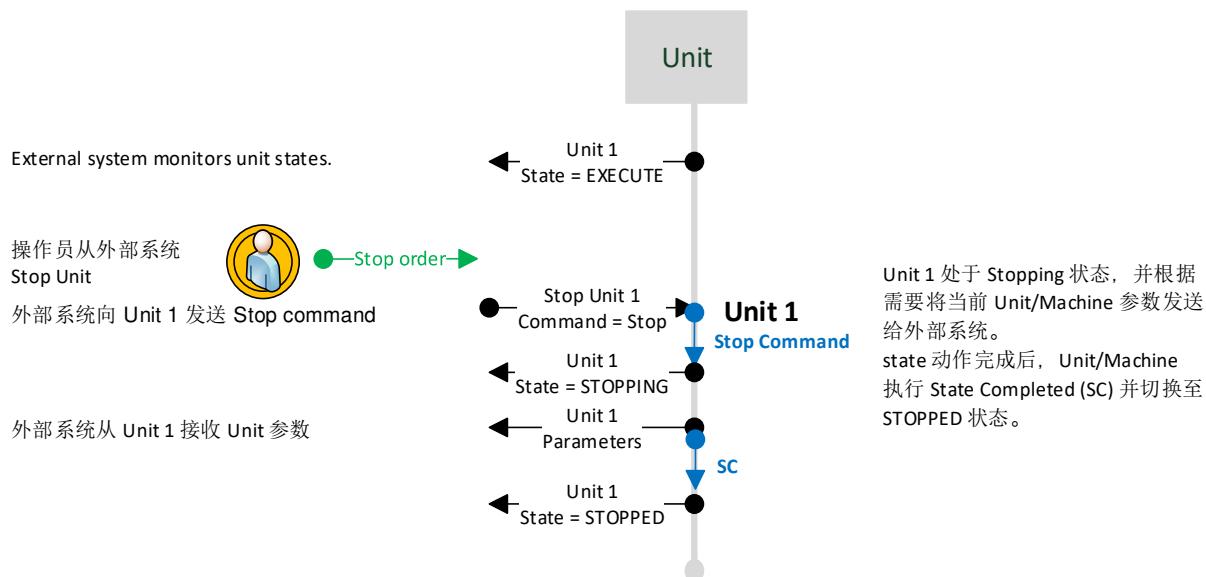


图 44：从外部系统完成

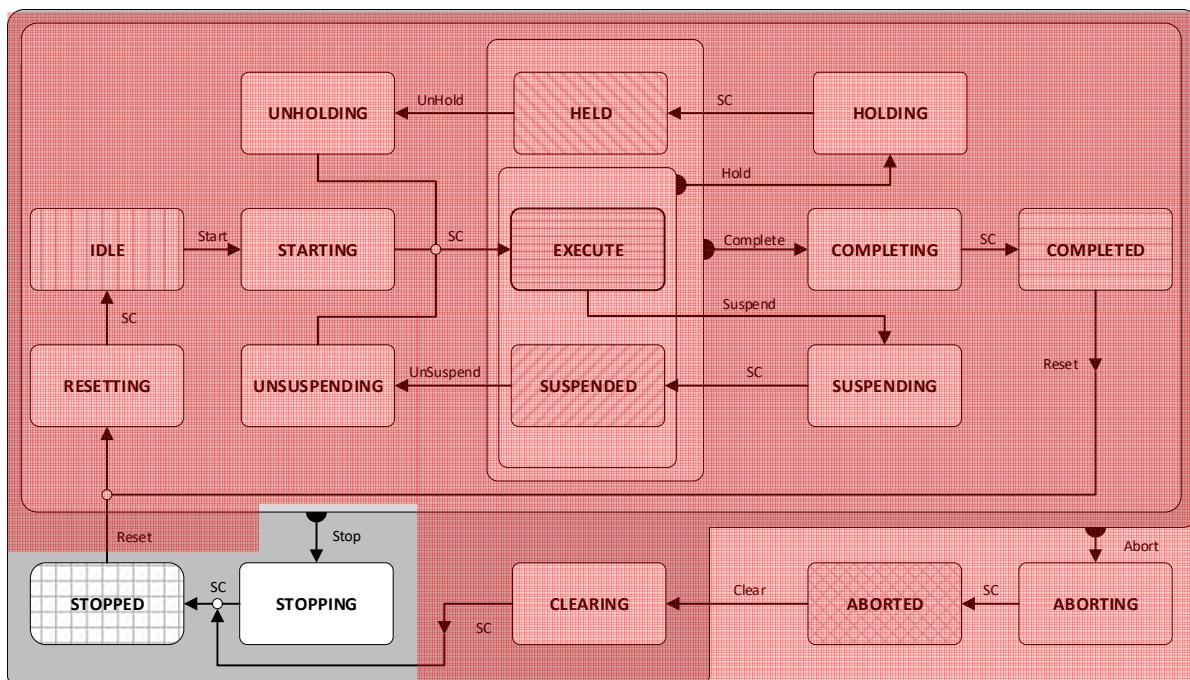


图 45：PackML state model 中状态聚焦于从外部系统停止订单

12.6 操作员中断和暂停使用 UNIT/MACHINE

操作员决定临时停机并将 Unit/Machine 置于 Suspended 状态。Unit/Machine Suspension 通常在需要短暂停歇或开会时使用。

外部系统也可以 Suspend Unit。

提示: Unit/Machine 的机器供应商和最终用户需要确认单元中的 OEE 参数会用于何种目的。

单元停止的 StopReason 会直接在单元内生成，然后不经修改发送至外部系统。

此外还需定义产品应该留在单元内还是应该从单元内清空。

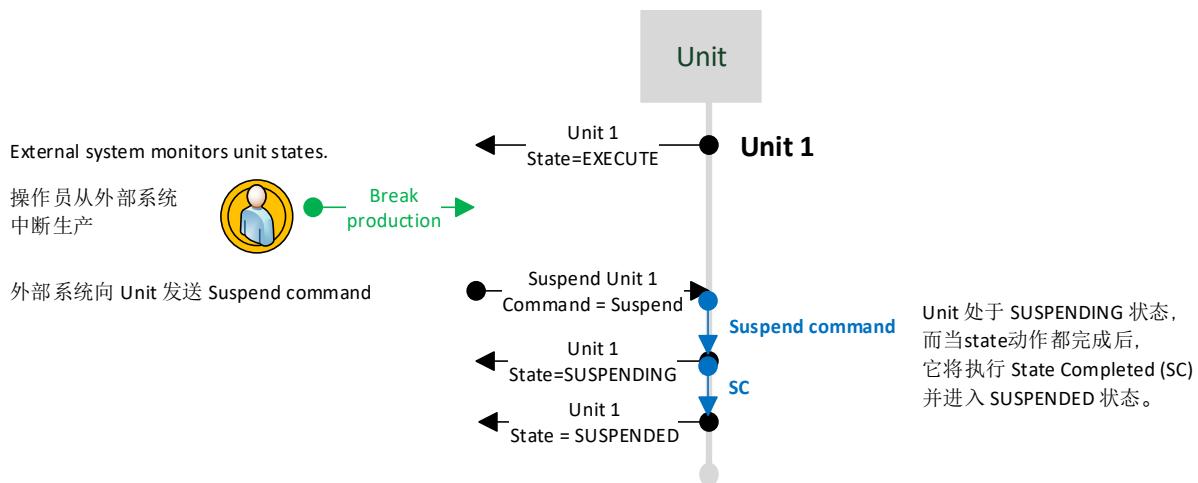


图 46: 操作员从外部系统中断生产

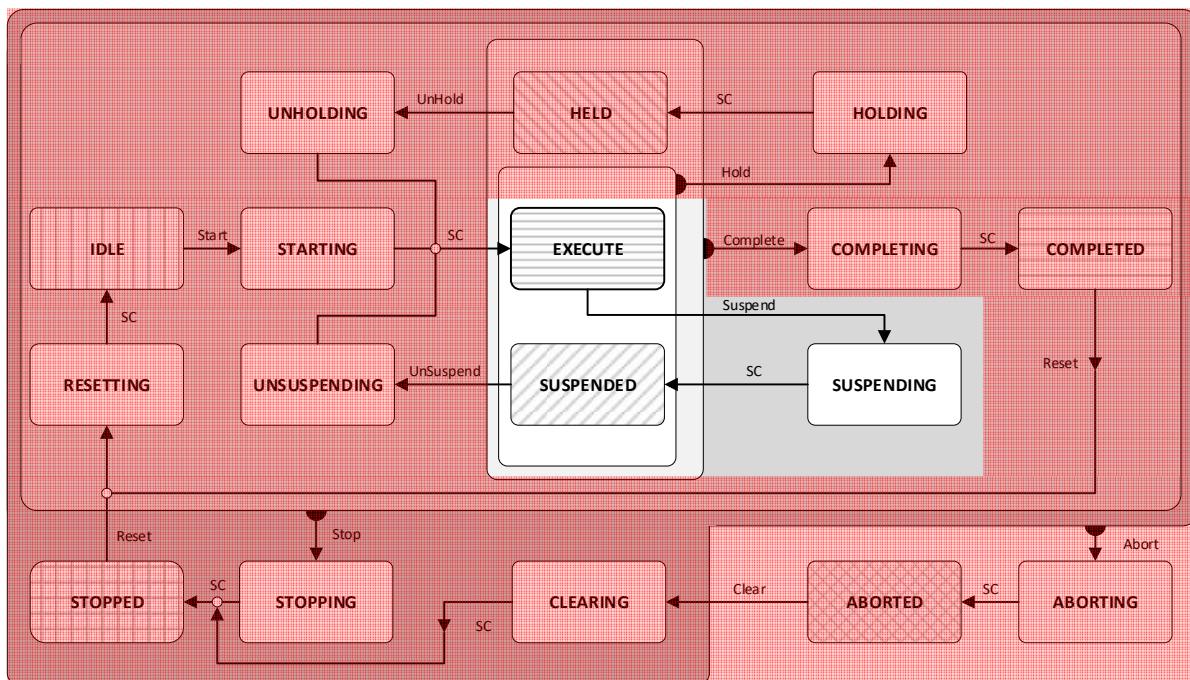


图 47：PackML state model 中状态聚焦于操作员从外部系统中断生产³⁵

12.7 中断后从外部系统恢复 UNIT/MACHINE

操作员 RESTART (UNSUSPEND) Unit/Machine。外部系统通过发送 Unsuspend 命令来将 Unit 解除挂起。Unit/Machine 将进入 EXECUTE 状态。

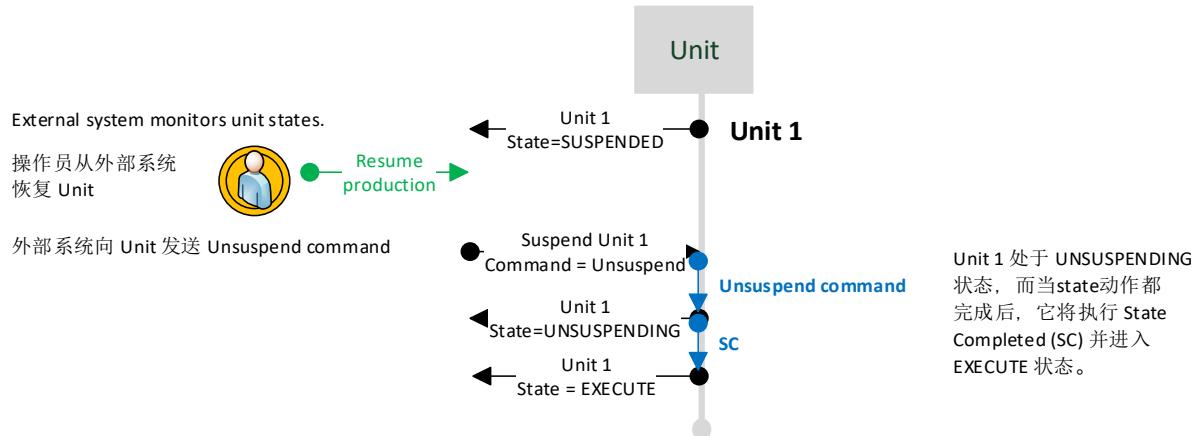


图 48：中断后恢复 Unit/Machine

³⁵ Unit/Machine 供应商专用 Machine State 模式可以为 Stopped。

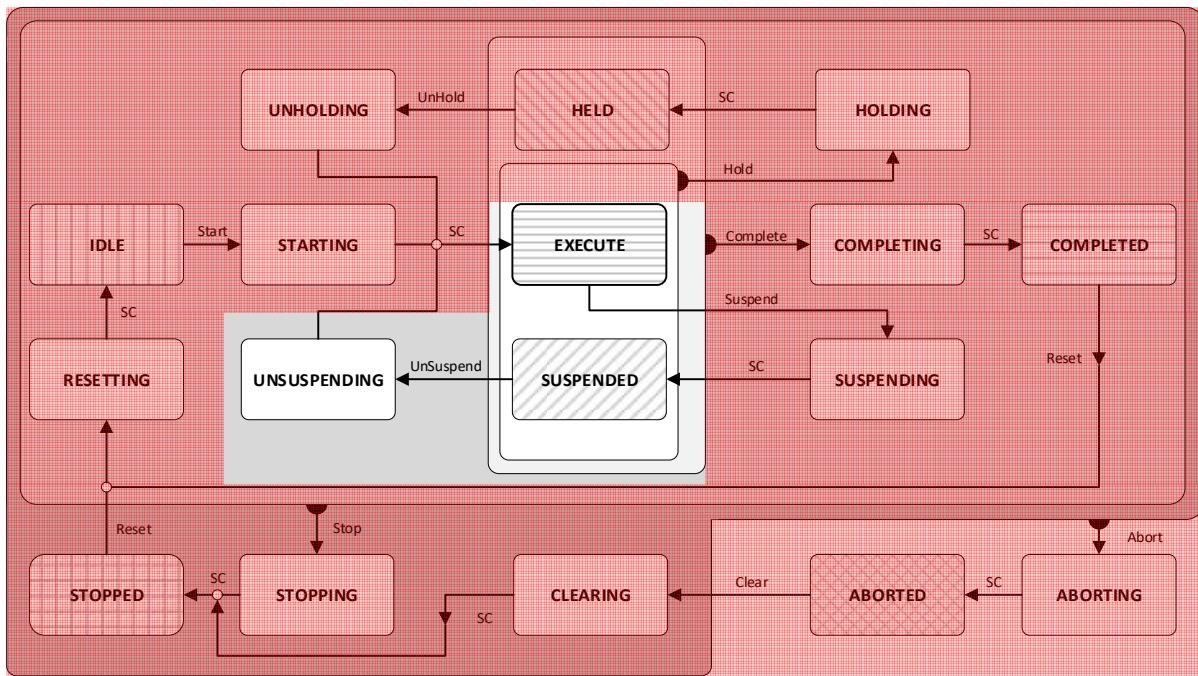


图 49: PackML state model 中状态聚焦于中断生产后从外部系统恢复 Unit/Machine

12.8 在 UNIT PANEL/HMI 停止 UNIT

操作员按 Unit/Machine 面板/HMI 上的 Stop 停止 Unit/Machine。当操作员希望启动 Unit 时，操作员需要 RESET Unit，使得 Unit 做好生产准备。下图显示了某个错误条件要求操作员停止 Unit。通常来说，当 Unit 上发生错误时，Unit 将进入 HELD 状态。举例来说，当 Unit 中使用的包装材料不正确时就会出现这种情况。

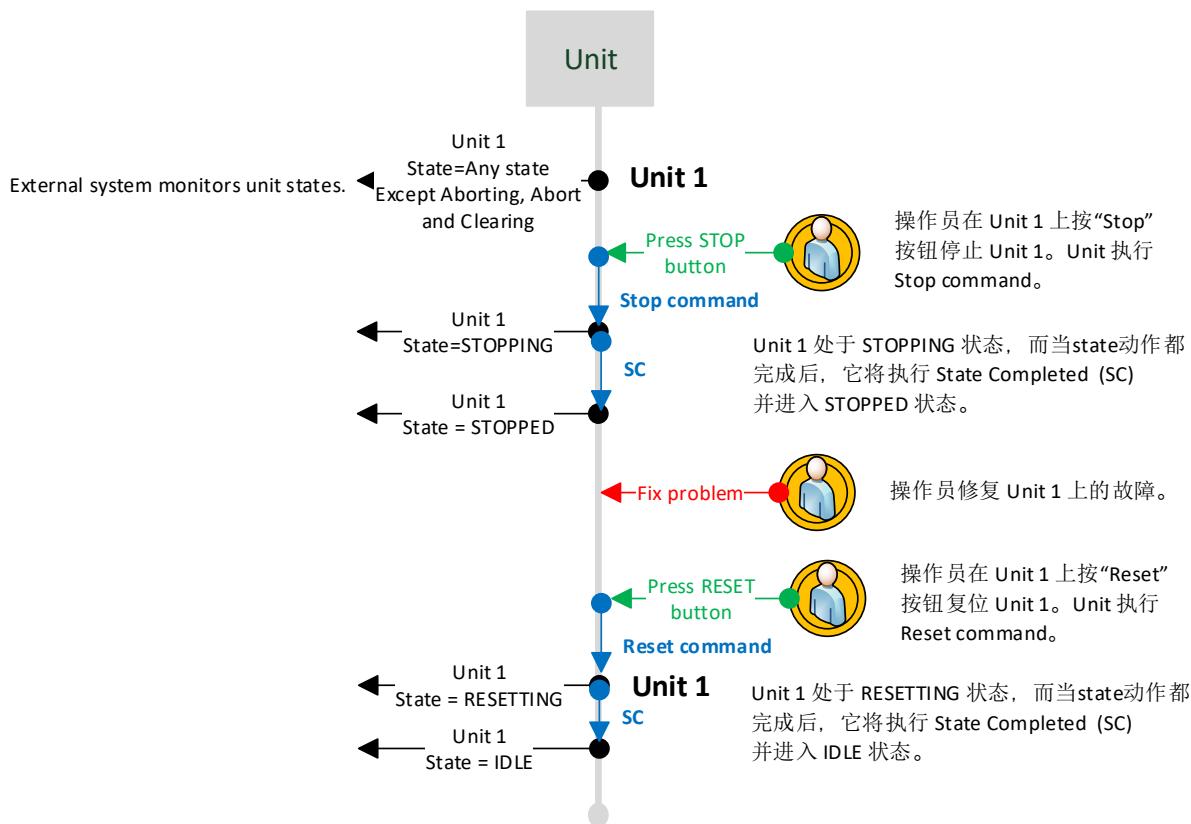


图 50：从 Unit 停止 Unit

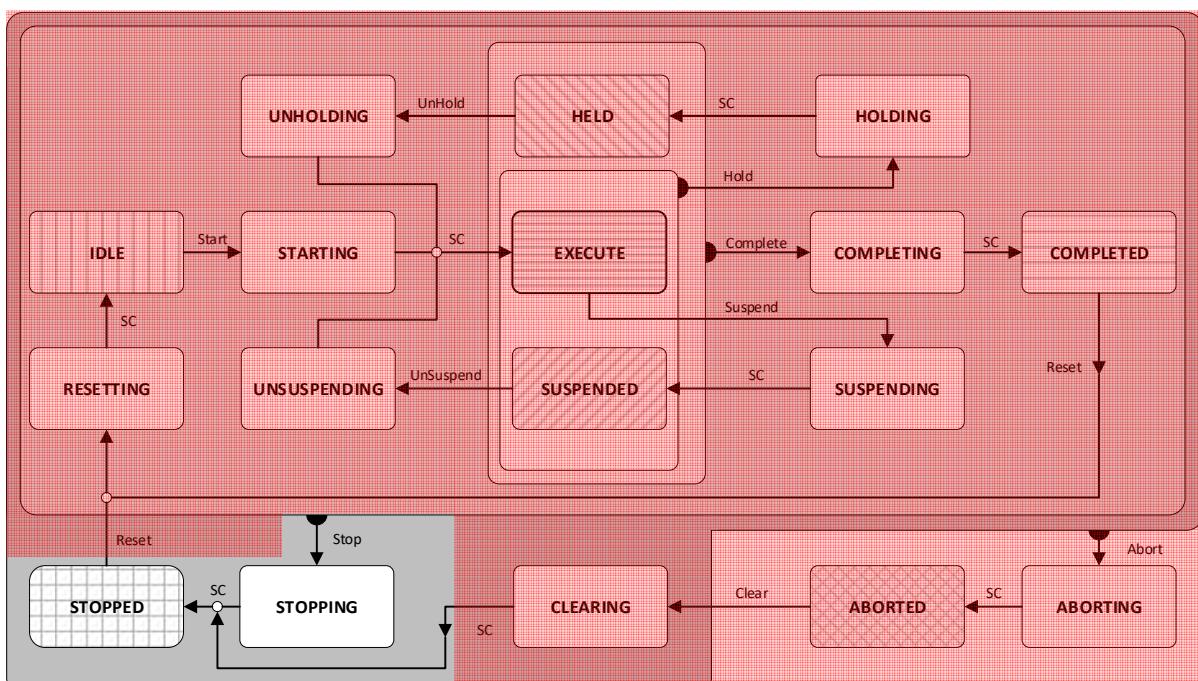


图 51：PackML state model 中状态聚焦于 Stop Unit

12.9 ABORT UNIT/MACHINE

操作员按 Unit/Machine 上的 E-stop 按钮。操作员修复 Unit/Machine 上可能存在的故障。不建议使用 E-stop 按钮来实现单元停机，因为相比停机或挂起，E-stop 并非受控停机。修复导致 E-stop 的故障后，操作员清除安全回路，并复位 Unit/Machine，最终让单元做好生产准备。

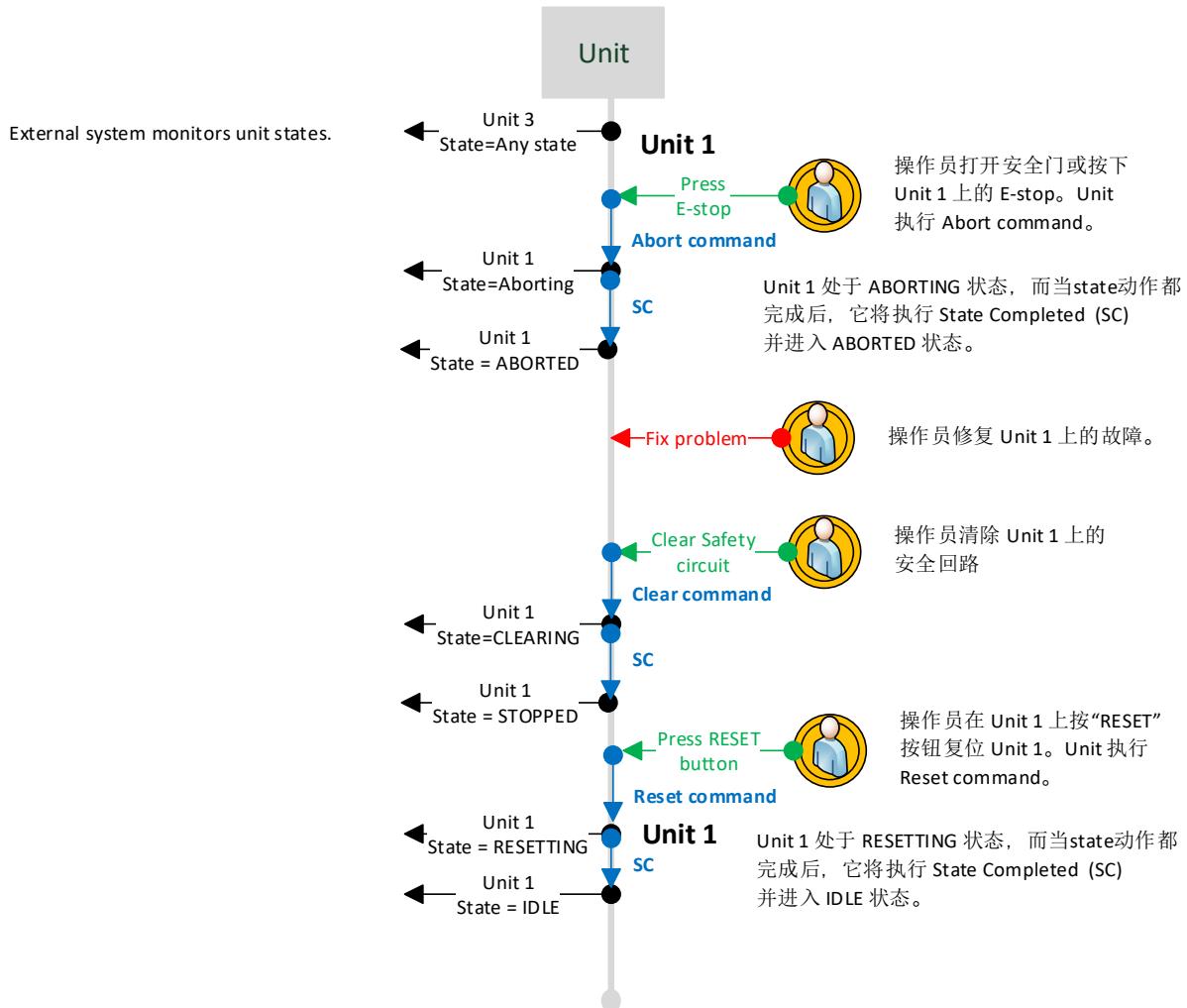


图 52: Abort Unit

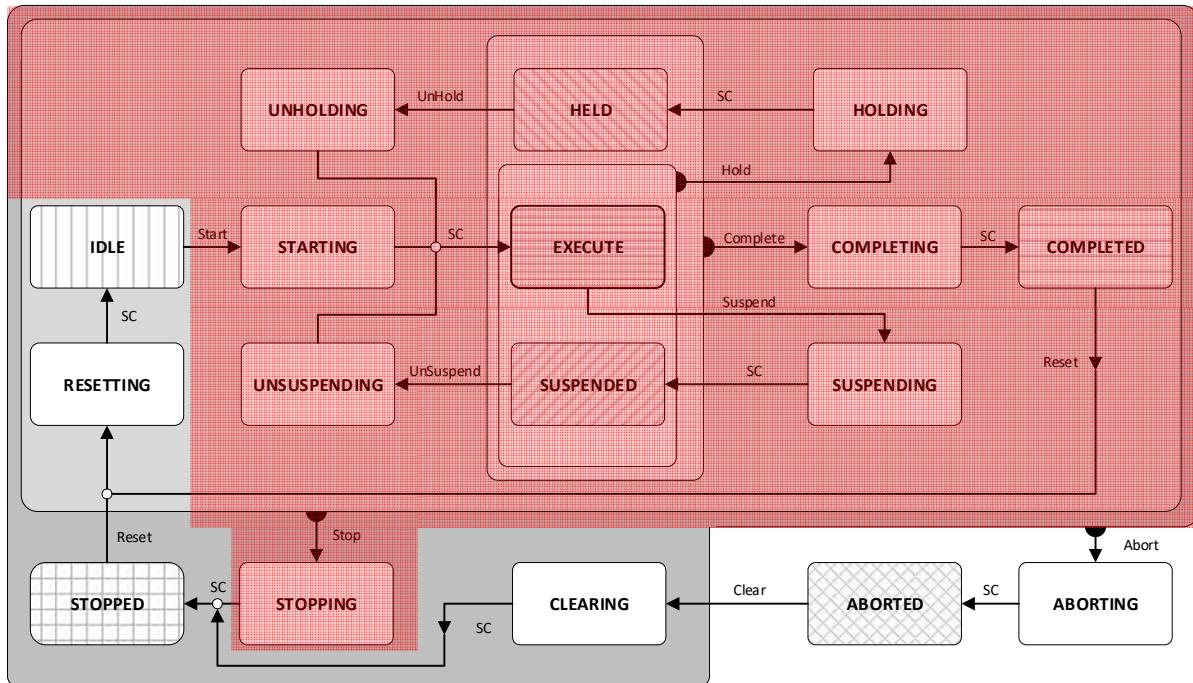


图 53: PackML state model 中状态聚焦于中止和重置生产

12.10 在 STOP 或 ABORT 后从 UNIT PANEL/HMI 重新启动 UNIT

操作员可以从 Unit Panel/HMI 重新启动单元。

下图说明了操作员从 Unit Panel/HMI 启动单元的情形。操作员按下单元上的 Start 按钮。恢复参数后，进入 EXECUTE 状态。

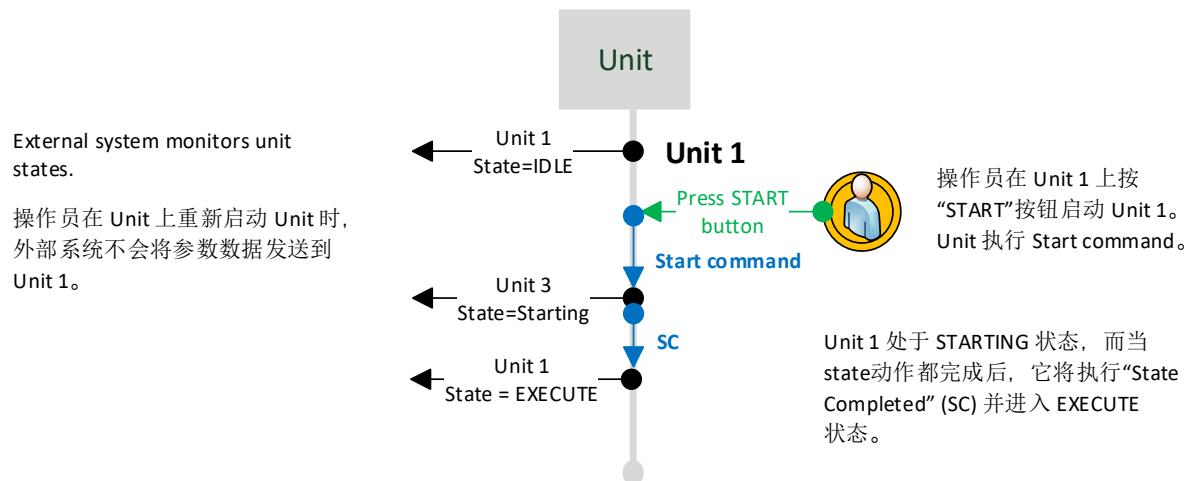


图 54: 在 Stop 或 Abort 后从 Unit 重新启动 Unit

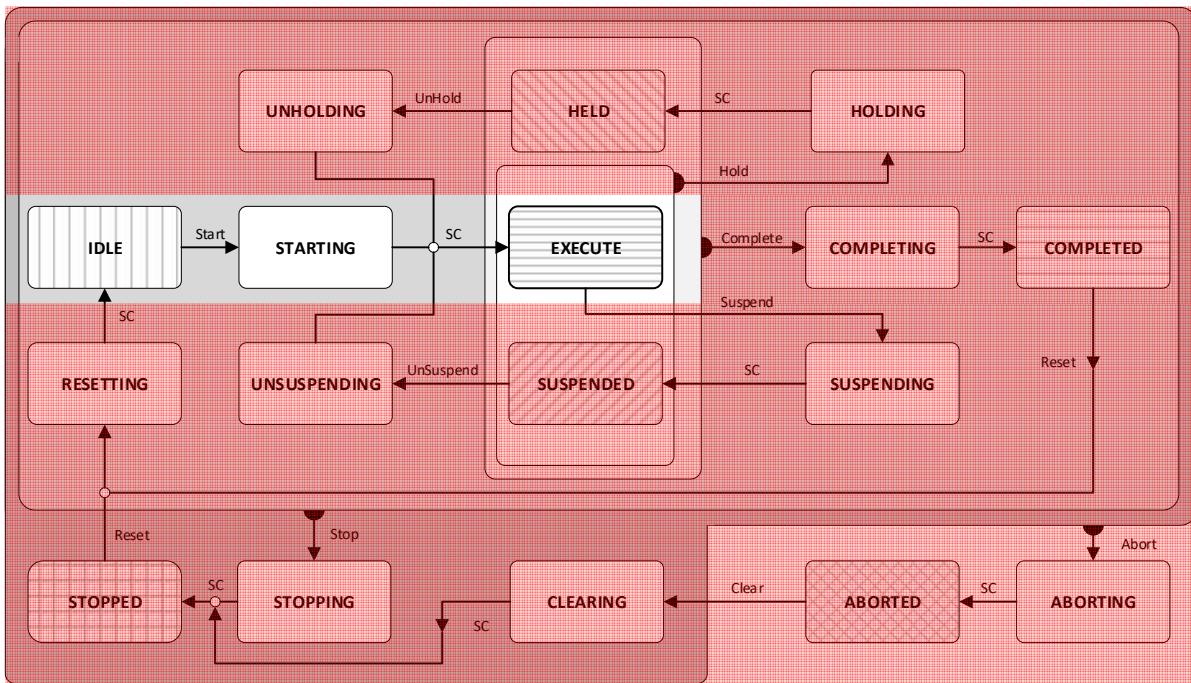


图 55: PackML state model 中状态聚焦于从 Unit Panel/HMI 重启 Unit。

12.11 在 STOP 或 ABORT 后从外部系统重新启动 UNIT

操作员也可以从外部系统重新启动 Unit。

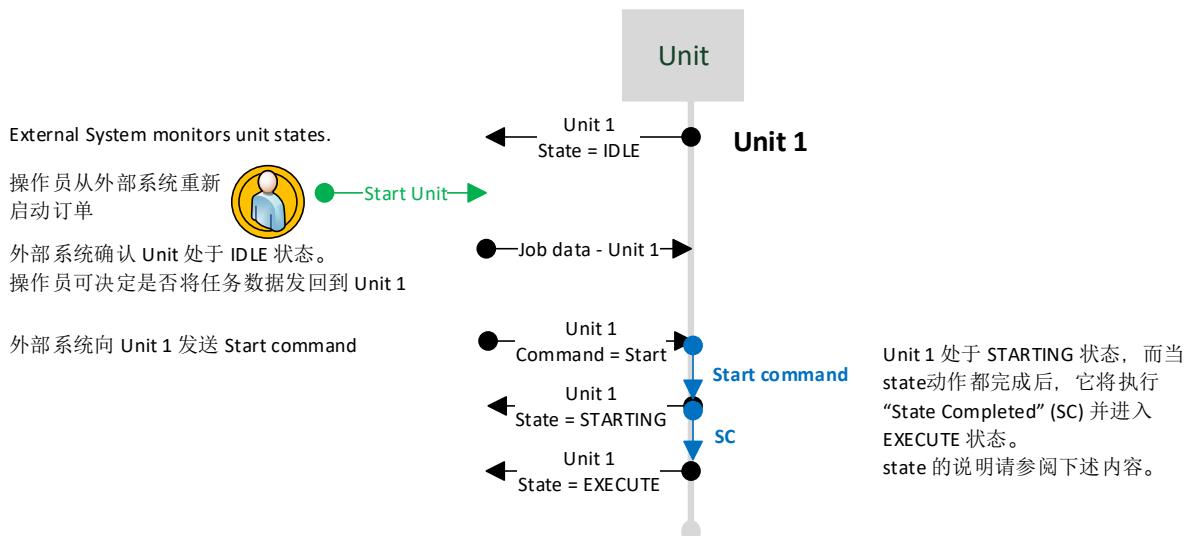


图 56: 在 Stop 或 Abort 后重新启动 Unit

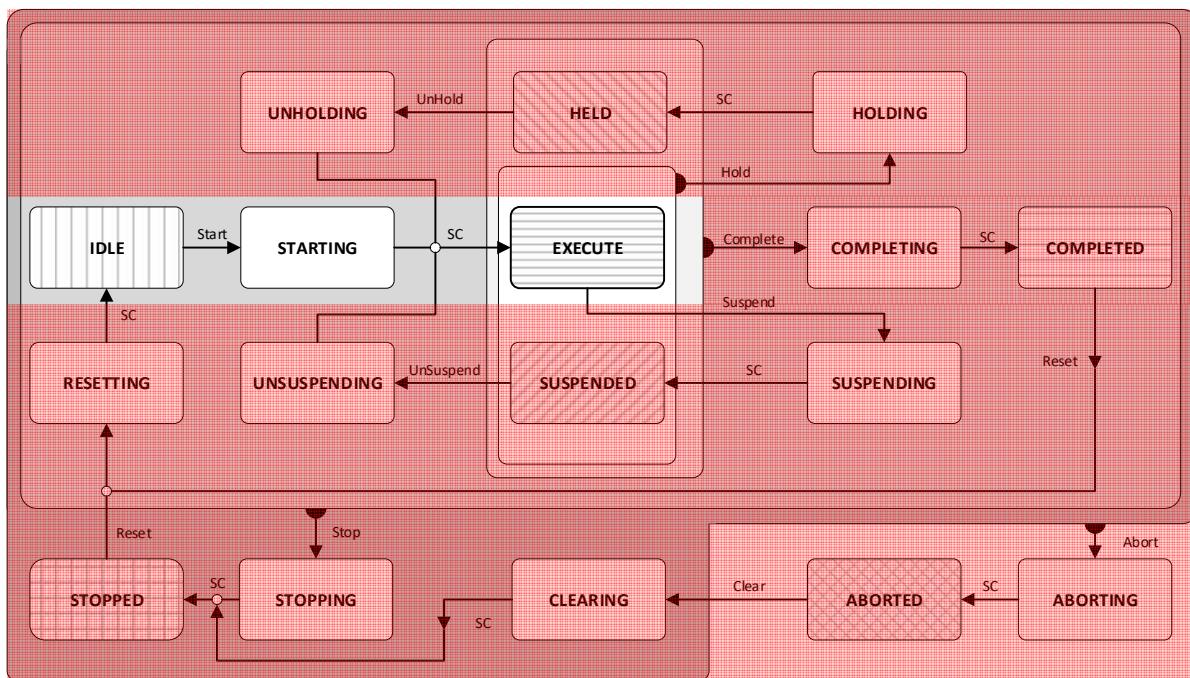


图 57: PackML state model 中状态聚焦于从外部监控系统重新启动 Unit。

12.12 UNIT 的报警和警告

alarm 的数据流参阅图 60。如果外部系统需要将 alarm 和 warning 数据用于 HMI 或 Historian 目的，则必须定义和描述从单元发送的所有 alarm 和 warning 数据。

如果发生 alarm 或 warning，操作员可以被引导至 Unit/Machine 了解情况。根据情况，Unit/Machine 必须执行某些操作。可以在 Event state management table 中定义要执行的操作。请参见第 12.12.1 节。如表 25 所示，Event state management table 是为了处理不同最终用户之间对 alarm 和 warning 的不同解释。最终用户可能并不都同意什么是 alarm 和 warning，因此，表格为机器供应商提供了灵活性，以符合当前最终用户的要求。

例如，最终用户可如下定义：

内部事件：

- Unit/Machine 的 alarm，例如驱动失败
- Unit/Machine 的 warning，例如包装材料缺失

将内部事件配置为以下几类：

- Safety related Event: Alarm
- Non Safety related events: Alarms and Warnings (technical)
- Material related Event: Alarm and Warning

12.12.1 PackML alarm 和 event state management table

我们可以制作一个表格，以方便机器供应商配置哪些类型的 alarm 和 event 会导致状态变化。可以用来记录某个机器何时应该切换到 PackML Interface State Model 中的哪个状态。而且，我们要特别关注当单元需要跳转至 HELD、STOPPED、SUSPENDED 和 ABORTED 状态时的情况，因为最终用户有时对何时应进入不同的状态有不同的解释和定义。

表 25: Alarm/Event state management table 示例

Alarm / Event	State change ID = Commands	Reaction
1	8	Abort
2	8	Abort
3	4	Held – Error on material
4	4	Held – Error on glue
5	3	Stop – Drive failure
6	3	Stop – Sensor failure
7	6	Suspend – Blocked
8	6	Suspend – Queue
9	20	Pop-up information on HMI – Executing state
10	20	Pop-up information on HMI – Executing state
....

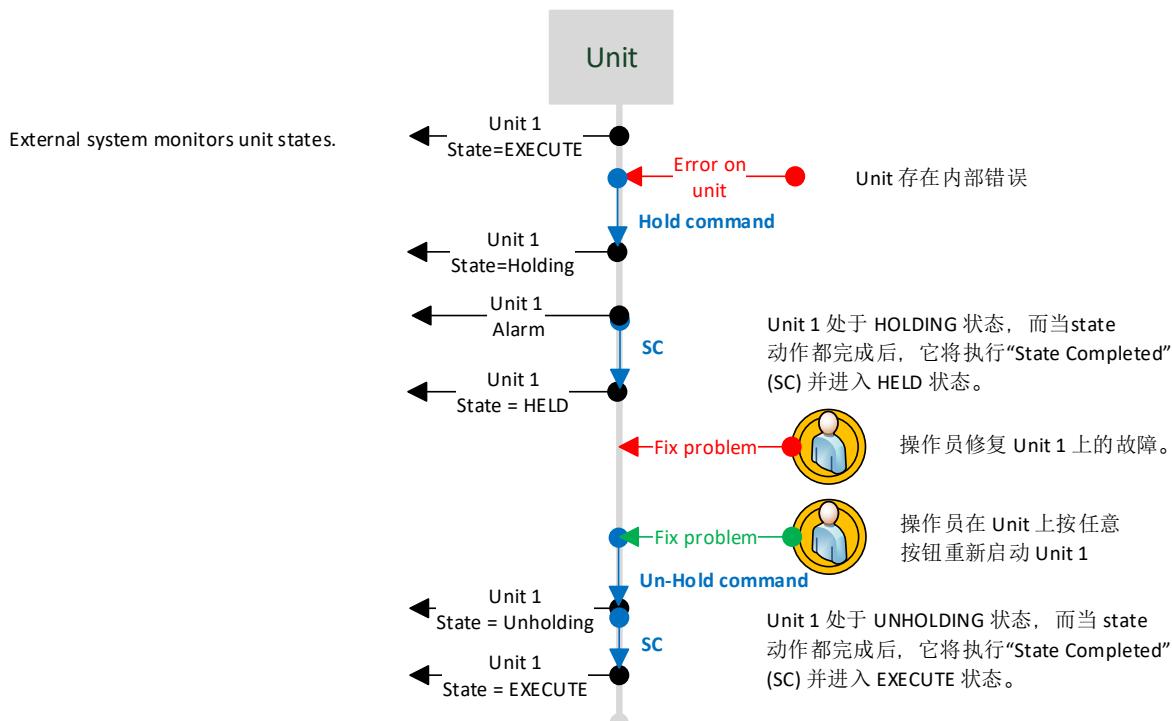


图 58: Unit 上的报警

图中显示了 warning 时的数据流。

提示: 对于从 Unit 发送的所有 warning 数据, 相关方需要定义外部系统是否需要使用这些 alarm 数据。



图 59: Unit 上的警告

注意: alarm 和 warning 可以在任何 PackML state 下发送, 因此需要考虑所有 PackML state。

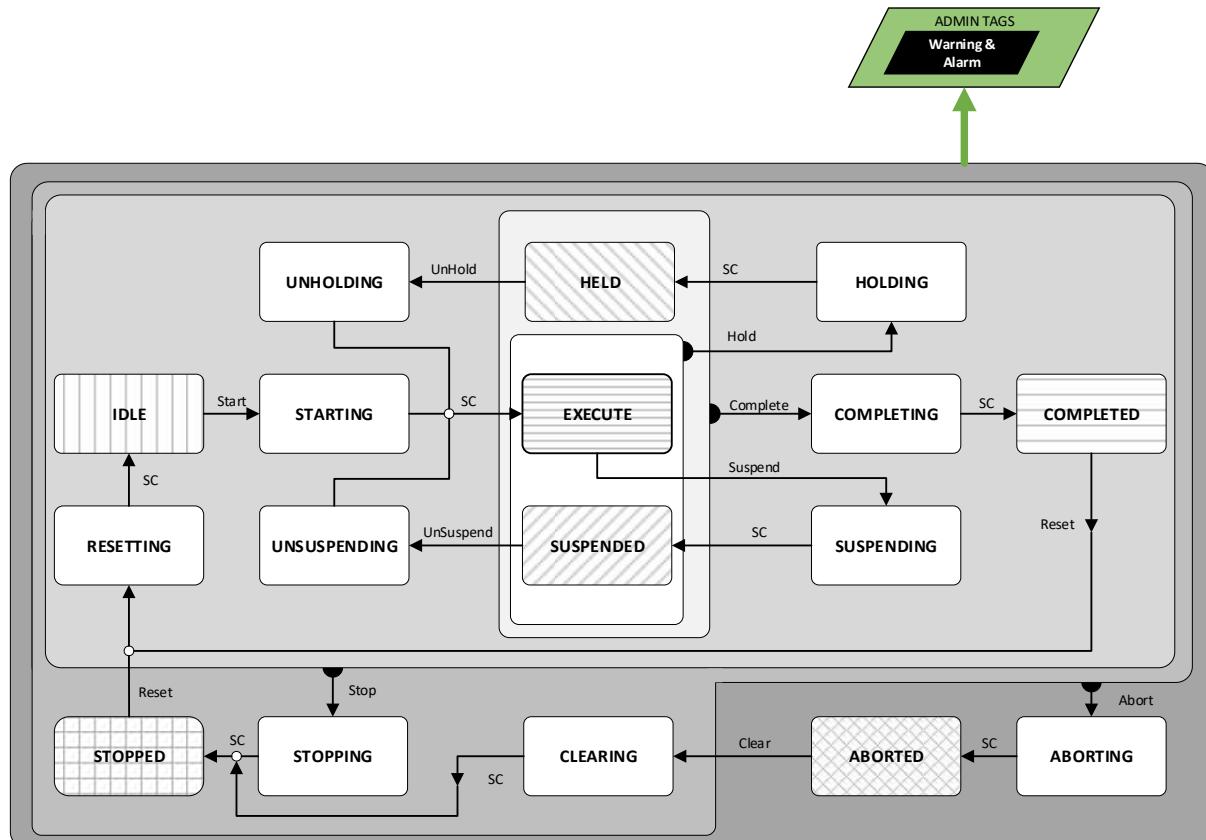


图 60: PackML state model 中状态聚焦于 alarm 和 warning 处理

12.13 背景任务：OEE

下面示例说明了用于 OEE 计算的原始数据的数据流。Unit 产生的数据会使用 StopReason.Value、ProdDefectiveCount、ProdProcessedCount 这些 PackTag 提供给外部系统。

在这种情况下，Unit/Machine 只以 PackTag 的形式发送 OEE 参数，随后由外部系统计算所有 OEE KPI 以执行 OEE 分析。

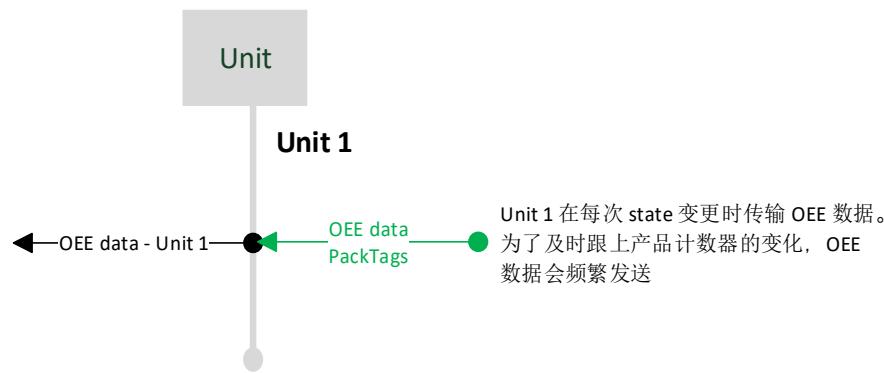


图 61：背景任务 - OEE

注意：OEE 数据可以在任何 PackML state 下发送，因此需要考虑所有 PackML state。

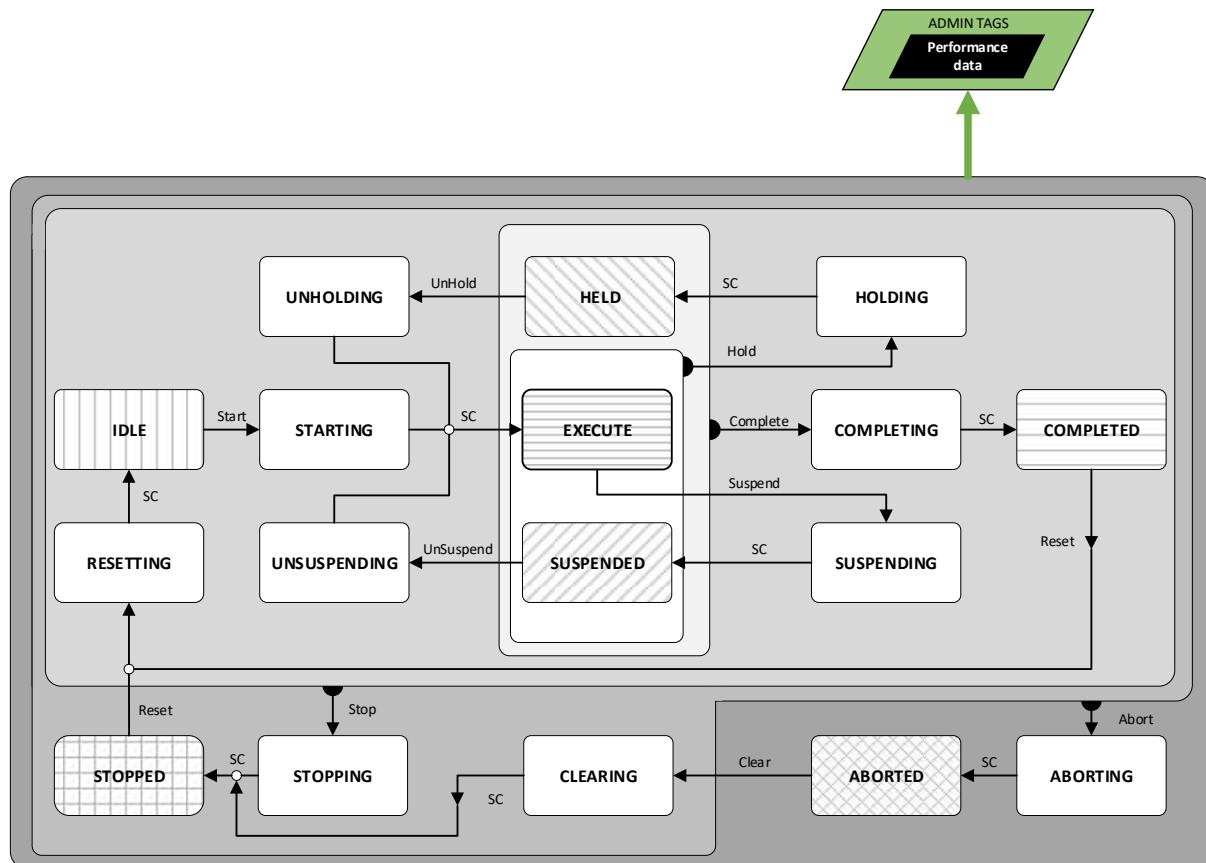


图 62：PackML state model 中的状态聚焦于背景任务：OEE

12.14 背景任务：MODE 和 STATE

在此示例中，PackML 的 State 和 Mode 可从一个 Unit 获得。外部系统可以收集来自 Unit 的 State 和 Mode，并可以使用它生成这台 Unit 或整个 Unit 生产线的概况。

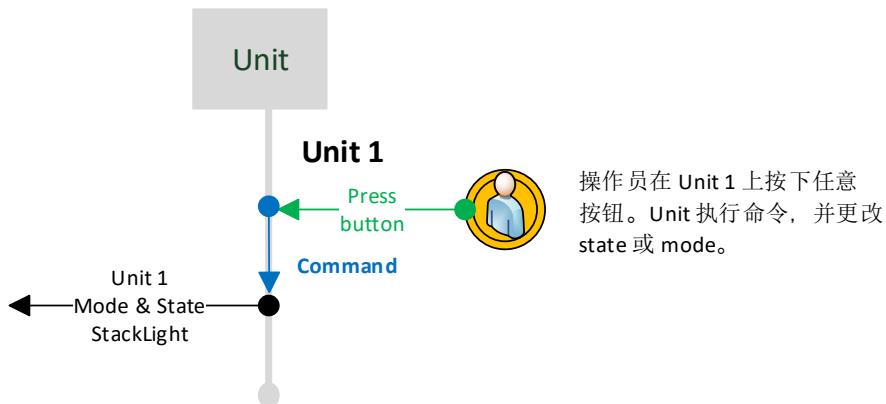


图 63: 变更 Mode 或 State, Unit 将发送 Mode 和 State 信息

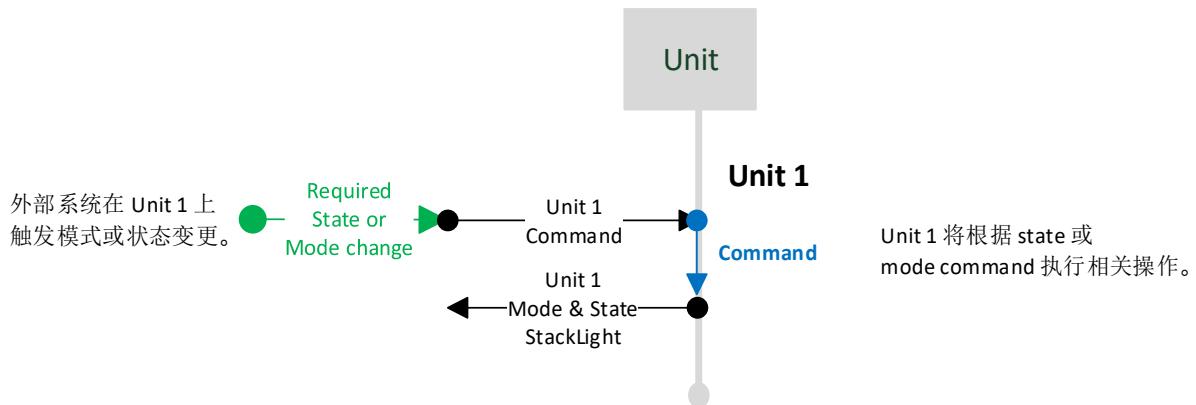


图 64: 发送 Mode 和 State 信息

注意: State 和 Mode 可以在任何 PackML state 下发送, 因此需要考虑所有 PackML state。

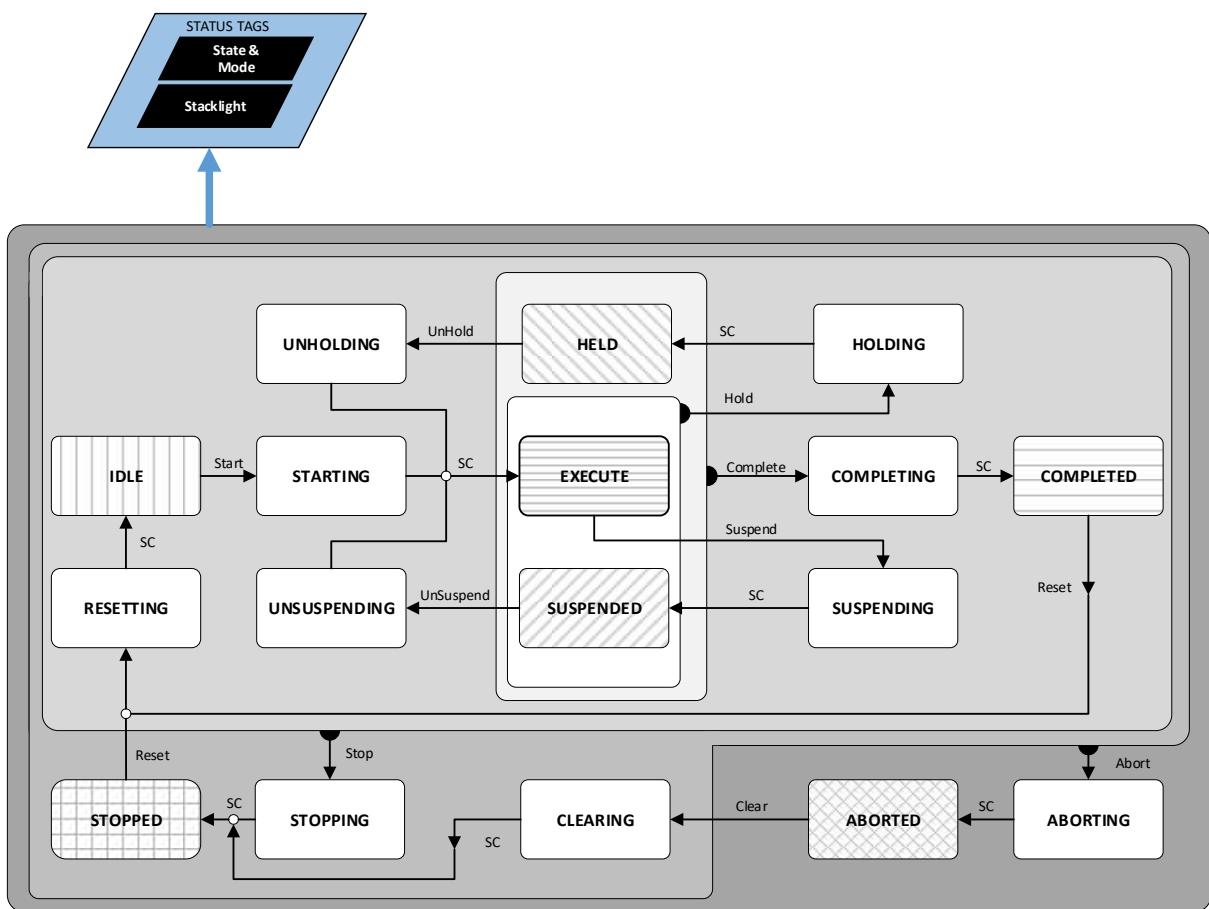


图 65: PackML state model 中状态聚焦于 Mode 和 State 报告

12.15 SUSPEND: 缺料或饱和

在此示例中，当产品 Queue guard 检测到 Unit 上出现缺料或饱和（blockage、queue）时，Unit 会进入 SUSPENDED 状态。



图 66：因缺料或饱和而暂停生产

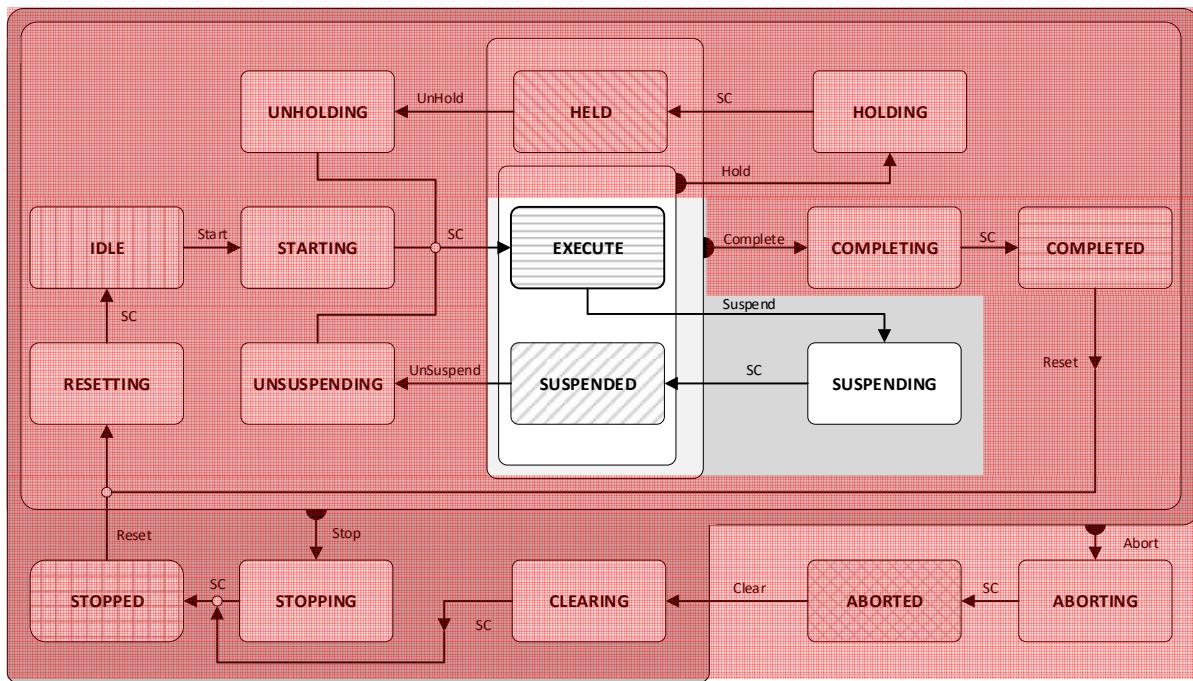


图 67：PackML state model 中的状态聚焦于暂停生产

12.16 UNSUSPEND

当 Queue guard 不再检测到单元上出现缺料或饱和（blockage、queue）时，单元会解除挂起返回 EXECUTE 状态。

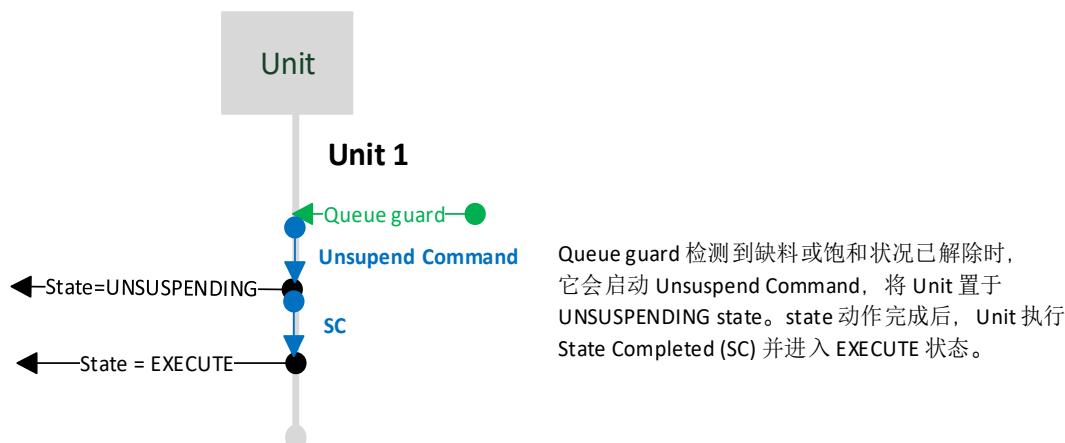


图 68：在暂停后重启生产

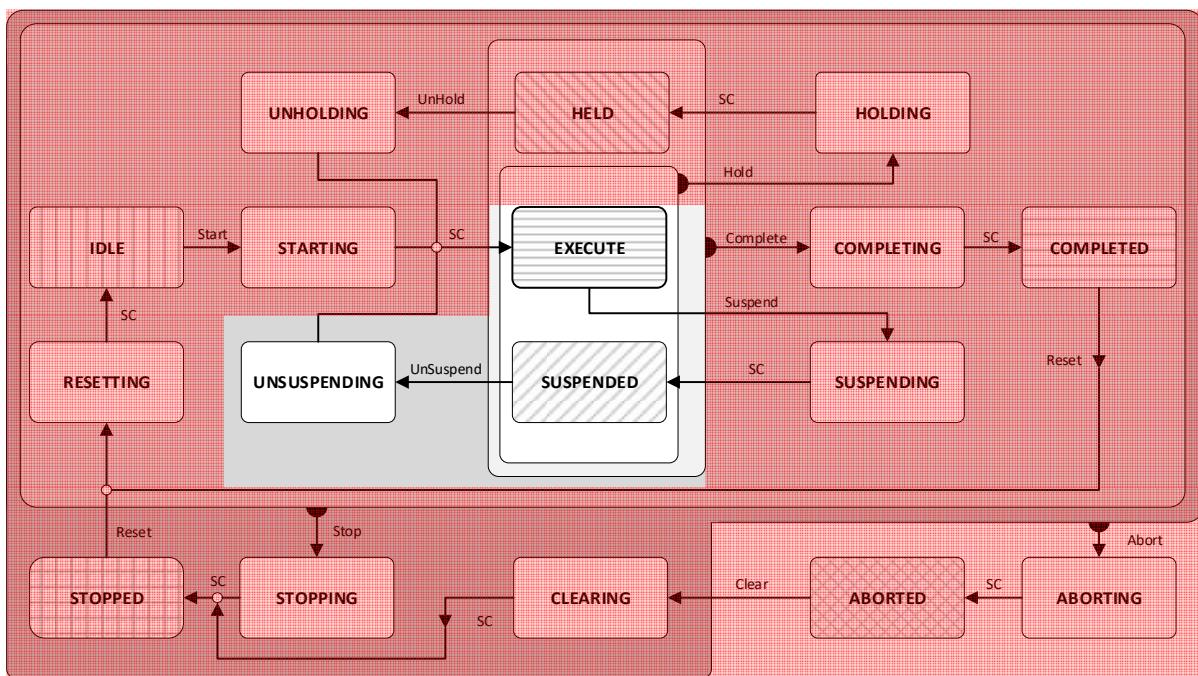


图 69：PackML state model 中的状态聚焦于 Unsuspending 生产。

13. 参考信息

13.1 定义和缩写

表 26：定义和缩写列表

缩写	内容	解释
FAT	Factory Acceptance Testing	交付前对 Unit/Machine 开展的测试。开展该测试是为了确保交付的 Unit/Machine 符合约定的要求。
FMEA	Failure Mode and Effect Analysis	FMEA 通常是系统可靠性研究的第一步。它涉及审查尽可能多的组件、组成和子系统，以便排查故障模式及其成因和影响。
HMI	Human Machine Interface	Human machine interface 是 Unit/Machine 的组成部分，用来处理人机交互。
I/O test	Input / Output test	I/O test 是在安装地开展的电气连接测试。
ISA	The International Society of Automation	ISA 是一个非营利性技术协会，其成员包括从事、研究或有兴趣了解工业自动化及其愿景的工程师、技术人员、企业家、教育工作者和学生。
OEE	Overall Equipment Effectiveness	OEE 经常与生产效果一起用作 Key performance indicator (KPI)，用来衡量生产成功与否。
OEM	Original Equipment Manufacturer	OEM 是生产的产品或组件被另一家公司购买并贴上购买公司品牌或公司名称的生产商。
OMAC	Organization for Machine Automation and Control	OMAC 是一个由自动化和制造业专业人士组成的组织，其宗旨是支持制造业对机器自动化和运营的需求。
PackML	Packaging Machine Language	PackML 的主要目的是为所有机器提供通用“界面外观”和操作一致性。
PackML GW	PackML Gateway	表示 PackML Interface State Manager 的网关，它会映射至特定机器 (Unit/Machine) 的现有 Machine State Manager。
PackML SDS	PackML Software Design Specification	Unit/Machine 的功能规范，它规定了 Unit/Machine 的行为，也指定了从 PackML 角度与 Unit/Machine 交互的接口。
PackTag	PackML data identifier	PackTag 是由 OMAC Packaging Workgroup 制定的数据元素，用来简化包装机器之间的数据交换。
PLC	Programmable Logic Controller	控制仪表、马达、阀门等零部件的计算机。
SAT	Site Acceptance Testing	在安装机器的生产点开展的 Unit/Machine 测试。开展该测试是为了确保交付的 Unit/Machine 符合约定的要求。

缩写	内容	解释
SCS	Supervisory Control System	<p>Manufacturing Execution System (MES) 将客户订单整合到 ERP 系统中，并为 SCADA 和 PLC 系统进行顺序编程的系统软件。</p> <p>Supervisory Control and Data Acquisition (SCADA) SCADA 系统通常会记录来自 Unit/Machine 的日志数据并监控其状态。SCADA 往往用作生产线上集成的一套 Unit/Machine 的 HMI。</p> <p>Line Controller (LC) Line Controller 可以是 Supervisory Control System 的一部分。Line Controller 用来控制生产线中的若干 Unit。</p> <p>Supervisory Control System 可以由 PLC 担任，也可以是现有 PLC、PC 或 Server 的一部分。</p>
TR 88.00.02	Technical Report	Technical Report 是关于在不同机器上实施 S88 的信息文档，它也是 PackML 的规范。

13.2 参考资料

表 27：参考文献列表

Ref. no.	Document
[S88-1]	ANSI/ISA-88.00.01-2010, Batch Control, Part 1: Models and Terminologies.
[TR88]	ANSI/ISA-TR88.00.02-2022, Machine and Unit/Machine States, An Implementation example of ANSI/ISA-88.00.01.
[S88-5]	ISA/DRAFT-88.00.05-2013, Working Draft 08, Batch Control, Part 5: Implementation Models & Terminologies for Modular Equipment Control.

13.3 参与编写本指南的人员名单

以下人士积极参与了本技术报告的编写工作:

表 28: 参与编写本技术报告的人员名单

Name	Surname	Company
Joachim	Bergmeyer	A+F Automation
Engelbert	Freiberger	Alpma
Jürgen	Wegscheider	Alpma
Arne	Svendsen	Arla Foods
Claus	Norup	Arla Foods
Enrico	Paolucci	B&R
Lazarus	Patsakas	B&R
Miodrag	Veselic	B&R
Sari	Germanos	B&R
Rob	Rawlyk	Beckhoff
Pascal	Witprächter	Bosch
Wolfgang	Martin	Bosch
Carl	Bostrom	Bosch Rexroth
Thomas	Haberkorn	Bosch Rexroth
Imbriaco	Giacomo	Coesia Engineering Center
Peder	Stocklund	Coloplast
Arthur C.	Smith	Coming
Emilian	Axinia	COPA-DATA
Van Dam	Maarten	COPA-DATA
Pierre	Henri	Danone Engineering Worldwide
Pasquali	Stefano	Gebo Cermex.
Fabian	Elsässer	Harro Höfliger
Manfred	Zoehrer	Haas Food Equipment
Karsten	Vollmer	KHS
Andreas	Gschrey	Krones
Christian	Rott	Krones
Uwe	Keiter	Lenze
Christoph	Riedmann	Lti-motion
Frank	Polky	Mars
Lee	Smith	Mettler Toledo
Heinz-Hermann	Bruemmer	Meurer
Jochen	Schohaus	Meurer
Malte	Schlüter	Mitsubishi
Claus	Botzenhardt	Multivac
Fabrice	Bertin	Nestlé
Geoff	Gooding	Nestlé

Name	Surname	Company
Jean-Claude	Pourchet	Nestlé
Kris	Matlock	Nestlé
Markus	Gehrke	Nestlé
Peter	Jarlvik	Norden Machinery
Carlos	Ruiz	Omron
Josep	Lario	Omron
Adal	Tecleab	P&G
Jason	Debruler	P&G
Jim	Blundy	P&G
Paul	Southman	P&G
Jay	Joyner	P&G
Mark	Ruberg	Pro Mach
Gord	Davison	Pro Mach
Gunther	Saelzer	Rockwell Automation
Christian	Chatel	Schneider Electric
Rainer	Beudert	Schneider Electric
Anthony	Esnault	Serac
Carsten	Nøkleby	SESAM-World
Michel-René	Kübler	SEW-EURODRIVE
Heiko	Soehner	Siemens
Kim	Meyer-Jacobsen	Siemens
Mike	Pieper	Siemens
Uwe	Zell	SIG Combibloc
Thomas	Rummel	Softing
Filippo	Sarafini	Tetra Pak
Heinrich	Iben	Tetra Pak
Kasper Korsholm	Christiansen	Velux
Ole	Riis	Velux
David	Tobon	Yaskawa

13.4 SUPPORT

The preparation of this version2 of the document is supported by **Mitsubishi Electric** and **SESAM-World**.

13.5 THE AUTHOR

Carsten Nøkleby is a senior consultant with focus on Automation and Production IT. The consulting jobs are done for customers like Beiersdorf, Arla Foods, Tetra Pak, Novo Nordisk, Nestlé, Grundfos, Velux, Baader & Linco, Danish Hospitals and many more. The consulting is related to strategic and tactical issues related to the introduction computers and new business processes on the factory floor. The technical systems handled by Carsten Nøkleby are PLC, SCADA, DCS and MES.

Carsten Nøkleby is responsible for running the Industrial networks SESAM with approx. 3.000 members. The network has its focus on distributing knowledge between the members. The main focus is on Automatization and Digitalization of manufacturing processes.

Carsten Nøkleby made a Ph.D.Thesis in 1992 at the Technical University of Denmark. The title was “Structure and Information for a flexible assembly cell – Issues in the analysis, design and implementation of virtual manufacturing devices for remote controlled application processes” .

The visionary

I have always worked with the Smart Industry applications. From the old days we called it Computer Integrated Manufacturing during my Ph.D. What we see today is more emphasis on integration of people, machinery, equipment and products. The internet is here and IoE is the future. The potential of combining existing and new technology with business processes and culture will make the change in the manufacturing environment.

14. VERSION HISTORY

Table 29 Version log

Date [YYYY-MM-DD]	Initials	Description	Version
2015-12-10	CAN	First version – Input from OMAC workshops in 2015	01A
2016-03-11	CAN	Second version – Document rewritten based on the definition of terminology and scope defined by the OMAC working group <ul style="list-style-type: none"> • Claus Norup, Arla Foods • Arne Svendsen, Arla Foods and OMAC • Kris Matlock (Nestlé) • Pascal Witprächtiger, Bosch Packaging • Uwe Keiter, Lenze and OMAC • Malte Schlüter, Mitsubishi Electric 	02A
2016-03-18	CAN	Sections added: <ul style="list-style-type: none"> • Understand and define a Unit • PackML Interface State Model • List of participants 	02B
2016-03-22	CAN	Review comments from Uwe Keiter and Claus Norup Section added: <ul style="list-style-type: none"> • Section 6.1 The syntax of the PackML State Model • Section 7 PackML Event State Manager • Section 8 PackML Control Command definitions • Section 9 PackML Interface State definitions 	02C
2016-04-08	CAN	<ul style="list-style-type: none"> • unit/machine control modes • PackTags • Overview of use case scenarios 	02D
2016-04-21	CAN	<ul style="list-style-type: none"> • Final version for review of whole reference group 	02E
2016-06-23	CAN	<ul style="list-style-type: none"> • All 262 received review comments are put into the document. • All figures are made neutral and the HMI part is presented in a separate section. • All the comments on the workshop 21st and 22nd June at Arla Foods Copenhagen. 	03A 03B 03K
2016-08-01	CAN	<ul style="list-style-type: none"> • The document has been review by Sofie Ambrosius, Arla Foods. 	03L
2016-08-03	CAN	<ul style="list-style-type: none"> • Minor details changed, e.g., participation list and mapping alarms to event triggers to PackML 	03M
2016-08-04	CAN	<ul style="list-style-type: none"> • Review and minor details are changed. 	03N
2016-08-05	CAN	<ul style="list-style-type: none"> • Final version for 2nd review 	03O 03P
2016-10-13	CAN	<ul style="list-style-type: none"> • Review comments are included. The comments are from the companies: Alpma, Arla Foods, BoschRexRoth, COPA-DATA, Corning, Krones, Mettler-Toledo, Nestlé, SEW-Eurodrive and Siemens. 	03Q ... 03Y
2016-11-02	CAN/DB	Linguistic review by Dennis Brandl, BR&L Consulting	1.00
2021-08-20	CAN	Updated version according to the updated ISA-TR88.00.02-2021. Major changes to state model & PackTags.	1.01
2021-12-08	CAN	Reviews of document is included	2.00
2022-09-02	CAN	Update according to final approved ISA-TR88.00.02-2021	2.01
2022-10-12	CAN	Updated according to the final agreed year for release of ISA-TR 88.00.02-2022.	2.02