第 2 章 系统建模

致知在格物 《礼记·大学》

系统建模是一种形式化的建模方法，规范了系统模型中的元素种类、以及元素之间关系，定义了可以使大家都明白其含义的一系列标识符。因此，系统建模需定义模型的语法并规定模型的形式是否合法的一系列规则。系统建模通常会提供一套建模工具，使用这个工具可以设计和实现系统的建模模型。

本章介绍智能嵌入式系统的基本建模方法和技术，包括用于规范离散控制建模的有限状态机 FSM、与数据融合的有限状态机 FSMD、描述连续与离散的混成自动机 HA，以及系统级建模语言 SysML。系统建模基本目的是使用形式化方法无歧义地描述物理场景，掌握物理世界的真实知识。

第 2.1 节 有限状态机

智能嵌入式系统重要功能之一就是离散事件控制，而控制的本质是系统由一个状态转换到另一个状态，如自动旋转式栅门在开和关状态之间进行转换，列车门自动控制系统也是在开和关状态之间转换，路口交通灯系统在红灯、黄灯和绿灯之间转换。这些具有有限个状态并进行状态间转换的智能嵌入式系统可以使用有限状态机进行建模。

2.1.1 有限状态机

有限状态机（Finite State Machine，FSM）也称有限状态自动机或有限自动机，是表示有限个状态以及在这些状态之间的转换等行为的形式化模型。有限状态机用于描写一个状态在何种条件下转换到另一个状态，描述状态控制流和转换流。

定义 2.1 （有限状态机）有限状态机形式化地定义为 4 元组（S, I, f, s_1），其中 S 是有限状态集{s_1, s_2, . . . , s_n}，其元素称为状态，I 是输入集{i_1, i_2, . . . , i_m}，其元素称为输入元素， f 是状态转移函数： S×I→S，一般是偏函数，即在一部分元素上有定义，s_1是初始状态。

一个有限状态机可用带权值的有向图表示。

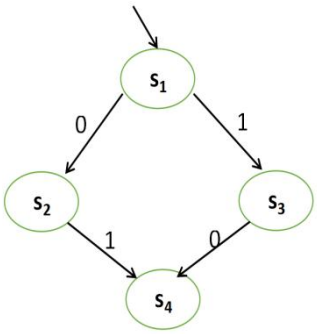


图 2-1 有限状态自动机

图 2-1 有限状态自动机含有四个状态 S_1，S_2，S_3，S_4，两个输入元素 0 和 1，有四个转移: $S_1 \rightarrow S_2$，$S_1 \rightarrow S_3$，$S_2 \rightarrow S_4$，$S_3 \rightarrow S_4$。这些状态转移与输入有关系，系统的输入触发了系统状态的转移，因此系统输入成了系统状态转移的触发因素。譬如，S_1 是初始状态，在 S_1 状态下， 当输入元素为 0 时，自动机从状态 S_1 转移到状态 S_2，而当输入元素为 1 时，则自动机从状态 S_1 转移到状态 S_3。

有限状态机的状态一般表示系统的功能，或者工作状态。如门开状态和关状态，交通灯处于红灯状态、绿灯状态或黄灯状态。输入元素是指有限状态机的可以输入元素，实现了系统与外界的交互，在不同的实际系统中表现形式是不一样的。 如自动门的输入是控制信号，表示是开还是关信号。交通灯的输入也是控制信号，表示红灯亮还是绿灯亮。状态转移函数规定了状态在输入元素情况下的转移情况。如自动门当前状态是关闭的，当输入是开门信号时，自动门状态转移到开的状态。交通灯当前是红灯亮状态，但系统输入是绿灯信号时，系统转移到绿灯亮状态。

2.1.2 有限状态机建模例子

自动门是指在信号控制下能自动开和关的门。在现代日常生活中经常遇到，如高铁的车厢门、电梯门、地铁站出入口的栅门。

例 2.1 自动旋转式栅门（Turnstile）

旋转式栅门是一个由三个齐腰高旋转柄组成的门，其中一个旋转柄在进道口。旋转式栅门一般安装在地铁站出入口等公共场所人行道，控制行人的进出。

初始时这旋转柄是锁住的，挡住进口，阻止行人通过。当交通卡（或投币）在旋转式栅门上的刷卡机扫描（或投币）成功时，这旋转柄就解锁了，允许一个

行人通过。在行人通过后，旋转柄又锁住了。

解：该系统用有限状态机可表示为：

状态集 S：{锁住，解锁}

输入集 I：{刷卡，通过}

状态转移函数 f：S×I→S：

（锁住，刷卡）→解锁

（解锁，通过）→锁住

初始状态：锁住。

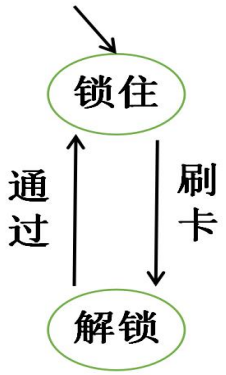


图 2-2 旋转式栅门 FSM

例 2.2 制热空调

制热空调是最简单一款空调，只能制热不能制冷，而且一旦加热就不能停止，除非关闭空调。

解：使用有限状态机可表示为：

状态集 S: {加热，关闭}

输入集 I: {开，关}

转移函数 f: S×I→S：

(关闭，开)→加热

(加热，关)→关闭

初始状态: 关闭。

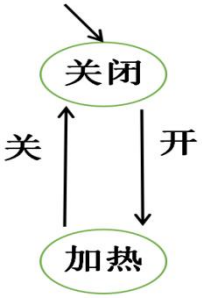


图 2-3 制热空调 FSM

例 2.3 餐巾纸售货机

系统功能描述：一款餐巾纸售货机，只接受 5 角和 1 元硬币，1 包餐巾纸价格为 1.5 元。

解：使用有限状态机可表示为：

状态集 S：{0 角，5 角，10 角，15 角，20 角}

初始状态：0 角

输入集 I: {5 角，10 角}。

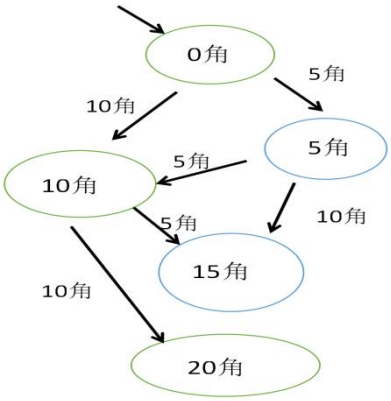


图 2-4 餐巾纸售货机 FSM

第 2.2 节 输入输出有限状态机

自动售货机是日常所见的智能产品，依据售货机上的产品价格进行自动售货。自动售货机除了拥有接受付款功能外，还有自动出货功能。对于自动售货机使用有限状态机进行建模明显是不够的， 因此需要改造有限状态机使之能对出货功能进行建模。

2.2.1 输入输出有限状态机

在有限状态机基础上，增加输出集和输出函数，形成输入输出有限状态机，也称 IO（Input Output）自动机。IO 自动机应用于与环境交互或者自适应智能系统的建模。而由于输出函数不同又将这种自动机分为两类：Moore 型自动机和 Mealy 型自动机。

定义 2.2 （输入输出有限状态机）输入输出有限状态机形式化地定义为一个 5 元组：（S，I，O，f，h），其中 S 是有限状态集$\{s_1, s_2, \ldots, s_n\}$；I 是输入集$\{i_1, i_2, \ldots, i_m\}$；O 是输出集$\{o_1, o_2, \ldots, o_k\}$；f 是状态转移函数：$S \times I \rightarrow S$；h 是输出函数。

根据输出函数 h 的不同，输入输出有限状态机可以分为 Moore 型自动机和 Mealy 型自动机两类[11]。

Moore 型自动机是基于状态的，其输出函数 h: $S \rightarrow O$。Mealy 型自动机则是基于状态和输入的，其输出函数 h: $S \times I \rightarrow O$。

Moore 型和 Mealy 型自动机分别如图 2-5(a)和(b)所示。

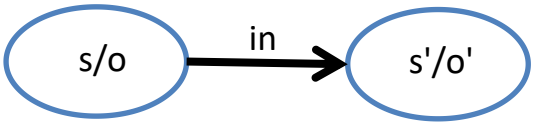


图 2-5(a) Moore 型自动机

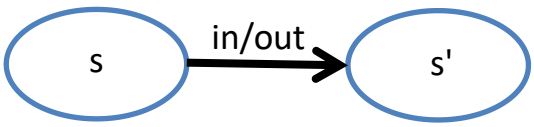


图 2-5(b) Mealy 型自动机

在 Moore 型自动机中，状态 s 在输入 in 之后转移到状态 s'，状态 s’下输出 o'；在 Mealy 型自动机中，状态 s 在输入 in 之后输出 out，同时转移到状态 s'。因此，它们不仅表现形式不同，在转移和输出顺序也是不同的。后面例子表明它们表现的方便性和复杂性更是不同的。

2.2.2 输入输出有限状态机建模

在 2.1.2 节中几个例子是使用有限状态机进行建模的，规范了状态的转换条件，但缺少输出功能。现在增加输出功能，更接近于实际控制系统。

例 2.4 自动旋转式栅门 turnstile 在原系统需求的基础上增加输出功能。分两种情况建模。

解：情形 1。在刷卡成功后解锁同时显示“请通过”，当机器锁住时显示“请刷卡”。该情形使用 Moore 型自动机建模。

状态集 S：{锁住，解锁}

输入集 I：{刷卡，通过}

输出集 O: {请通过，请刷卡}

状态转移函数 f：S×I→S

（锁住，刷卡）→解锁

（解锁，通过）→锁住；

输出函数 H: S→O

锁住→请刷卡，解锁→请通过。

初始状态：锁住。

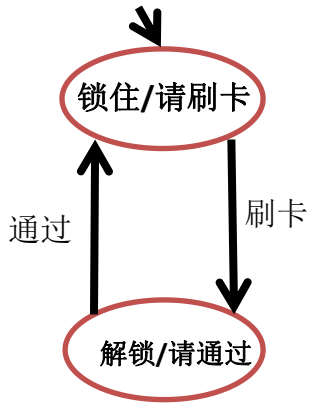


图 2-6 (a)栅门 Moore 型自动机

情形 2：若使用 Mealy 型自动机建模，输出元素需要做调整，才能更符合实际情况。

输出集 O: {请通过，谢谢}，

输出函数 H: S×I→O

(锁住，刷卡)→请通过，

(解锁，通过)→谢谢。

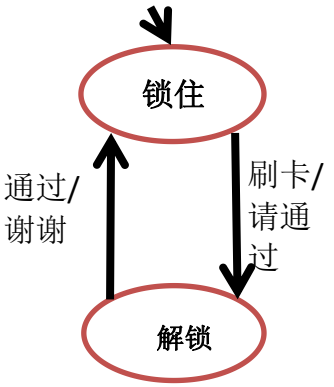


图 2-6 (b)栅门 Mealy 型自动机

例 2.5 餐巾纸售货机

一款餐巾纸售货机，只接受 5 角和 10 角，1 包餐巾纸价格为 15 角，有找零功能。

解：使用 Moore 型自动机建模

- 状态集 S：{0 角，5 角，10 角，10 角，20 角}
- 输入集 I:{ 5 角，10 角，X}　//X 表示无输入

- 输出集 O:{0 包，1 包}
- 转移函数 f: S×I→S

  (0 角，5 角)→5 角|

  (0 角，10 角)→10 角|

  (5 角，5 角)→10 角

  (5 角，10 角)→15 角|

  (10 角，5 角)→15 角

  (10 角，10 角)→20 角|

  (15 角，X)→0 角|

  (20 角，X)→0 角

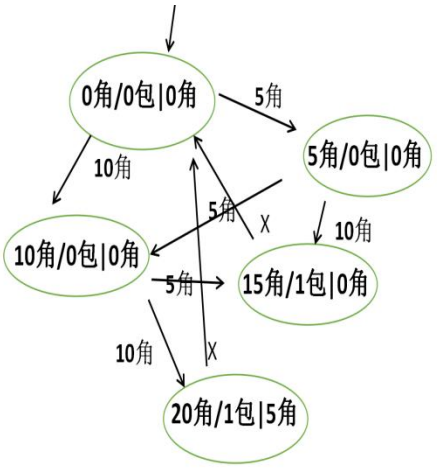


图 2-7 售货机 Moore 型自动机

- 输出函数 h: S→O

  15 角→1 包|0 角，20 角→1 包|5 角，

  5 角→0 包|0 角，10 角→0 包|0 角，

  0 角→0 包|0 角，

其中符号“1 包|5 角”表示同时输出 1 包和 5 角。

使用 Mealy 型自动机建模

- 状态集 S：{0 角，5 角，10 角}
- 输入集 I: {5 角，10 角}
- 输出集 O:{0 包，1 包，0 角，5 角}
- 纸转移函数 f: S×I→S

  (0 角，5 角)→5 角|

  (0 角，10 角)→10 角|

  (5 角，5 角)→10 角

  (5 角，10 角)→0 角|

  (10 角，5 角)→0 角|

  (10 角，10 角)→0 角

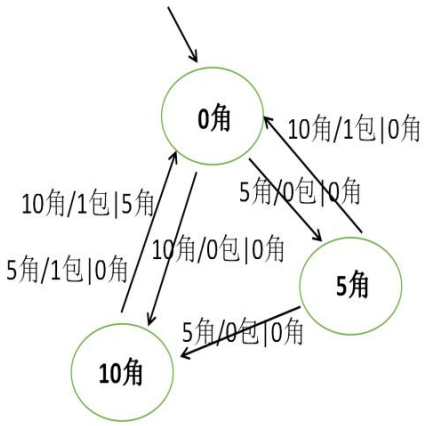


图 2-8 售货机 Mealy 型自动机

- 输出函数 h: S×I→O

  (5 角，5 角）→0 包|0 角，(5 角,10 角)→1 包|0 角，

  (10 角,5 角)→1 包|0 角，(10 角，10 角)→1 包|5 角。

例 2.6 时序检测器（Sequence）

一个时序检测器实现在接收连续 3 个 1 后输出 1， 其它情况输出 0。

解：状态集 S：{S0，S1，S2，S3}，其中 S0: 检测到零个 1， S1:检测到壹个 1，S2: 检测到贰个 1， S3: 检测到叁个 1。

输入集 I:{ 0,1}

输出集 O:{0,1}

转移函数 f: S×I→S

(S0，0)→S0，(S0，1)→S1，(S1，0)→S0，(S1，1)→S2,

(S2，0)→S0，(S2，1)→S3，(S3，0)→S0，(S3，1)→S3。

输出函数 h 要分成 Moore 型和 Mealy 型，

用 Moore 型自动机和 Mealy 型自动机建模分别如图 2-9 和 2-10 所示。

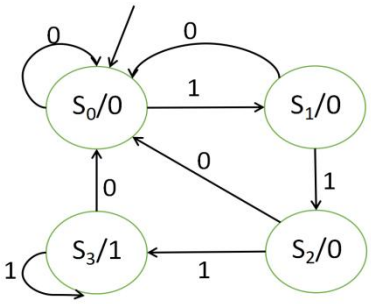


图 2-9 时序检测 Moore 自动机

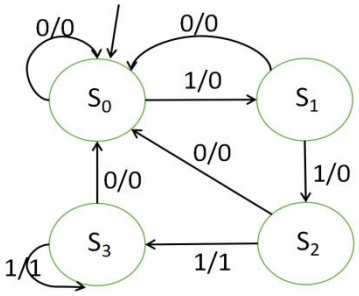


图 2-10 时序检测 Mealy 自动机

例 2.7 电梯控制[12] 设有一座有三层高的楼，装有电梯 1 部，每层楼都能到达，分别使用 Mealy 和 Moore 型自动机设计电梯控制系统模型。

解：Mealy 型自动机：

状态集：S={s1，s2，s3}表示楼层集， 如 s3 表示第 3 层楼。

输入集： I={r1，r2，r3} 表示要到达的楼层，如 r2 表示电梯要到达第二层楼。

输出集： O={d2，d1，n，u1，u2}表示方向和电梯要移动的楼层，如 d2 表示电梯

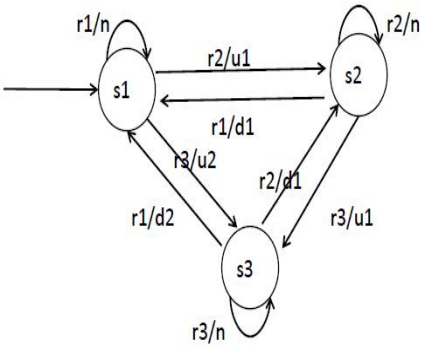


图 2-11 电梯 Mealy 型自动机

下降 2 层， u2 表示电梯上升 2 层， 而 n 表示电梯保持该楼层。

转移函数：f: S×I→S

(s1，r1)→s1， (s1，r2)→s2，(s1，r3)→s3，

(s2，r1)→s1，(s2，r2)→s2，(s2，r3)→s3，

(s3，r1)→s1，(s3，r2)→s2，(s3，r3)→s3。

输出函数：h: S×I→O

(s1，r1)→n，(s1，r2)→u1，(s1，r3)→u2，

(s2，r1)→d1，(s2，r2)→n，(s2，r3)→u1，

(s3，r1)→d2，(s3，r2)→d1，(s3，r3)→n。

Moore 型自动机

Moore 型自动机的输出函数只依赖于状态， 因此在同一个楼层，也有三种输出：上升 u、下降 d 和空闲 n。这样需要把每个楼层状态分成三个子状态，如 s1 分成 s11，s12，s13，以便同这三种输出进行组合。 结果 Moore 型自动机就有了 9 个状态，

状态集：S={s11，s12，s13，s21，s22，s23，s31，s32，s33}

输出集：O={d2，d1，n，u1，u2}

把 S 和 O 进行笛卡尔乘积，应该得到的集合 S×O 共有 45 个元素，即 45 个楼层和输出组成的组合状态，但实际上不需要这么多。如对于第一层来说， 输出只有空闲 n（保持在第一层），到达 1 层的 d1(从第二层到达第一层）和 d2（从第三层到达第一层）。对于第二层，输出有空闲 n（保持在第二层），上升到 2 层的 u1（从第一层上升到第二层）和下降到 2 层的 d1（从第三层下降到第二层）。而对于第三层，输出有空闲 n（保持在第三层），上升到 3 层的 u1(从第二层上升到第三层）和 u2（从第一层上升到第三层）。这样共有 9 个有用的组合：

X={(s11/d2)，(s12/d1)，(s13/n)，(s21/d1)，(s22/n)，(s23/u1)，(s31/n)， (s32/u1)，(s33/u2)}，得到输出函数 h: S→O，定义为: h(s11)=d2，h(s12)=d1，h(s13)=n；h(s21)=d1，h(s22)=n，h(s23)=u1； h(s31)=n，h(s32)=u1，h(s33)=u2。

输入集： I={r1，r2，r3}

转移函数：f: S×I→S 如图 2-12 所示。

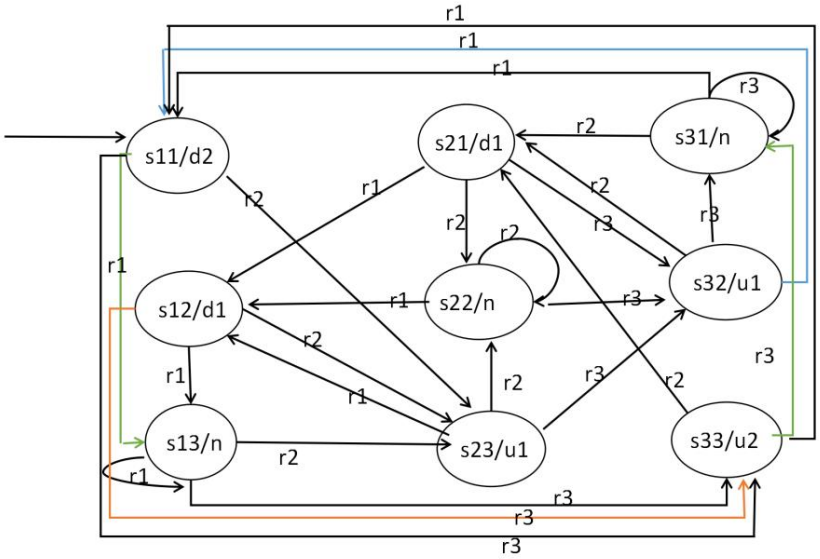


图 2-12 电梯 Moore 型自动机模型

第 2.3 节 数据有限状态机

第 2.2 节的电梯楼层转换自动机图有点复杂。另外，向汽车超速自动提醒系统，超速提醒依赖于汽车行驶的速度，再如，自动加温空调的开和关是依赖于房间的温度变化而转换的。因此需要介绍一种能简单处理依赖于系统数据变化的有限状态机模型：数据有限状态机。

2.3.1 数据流图(Dataflow Graph，DFG)

数据流图是最普遍的描述计算密集系统的方法。 数学公式可以自然地由一个有向图表示，其中节点代表计算或函数，边代表节点执行的顺序。

计算的数据流模型是基于异步和功能性原则的。 异步原则阐述了所有计算执行仅当待执行的计算是来源于可计算状态，而功能性原则阐述了所有计算行为是函数式的，没有任何的副作用。这样就可以得到任何可执行的运算可顺序或者并行地执行。

定义 2.3 （数据流图[12]）数据流图形式地定义为四元组：$\langle N, A, V, v^0\rangle$，其中 $N=\{n_1, n_2, \ldots, n_M\}$ 是节点集合，$A=\{a_1, a_2, \ldots, a_L\} \subseteq N \times N$ 是两节点间边的集合，集合 $V=\{\langle v_1, v_2, \ldots, v_L\rangle \mid v_i \in V_i, i=1 \ldots L\} \subseteq V_1 \times V_2 \times \ldots \times V_L$ 表示由每

个边值与特殊元素⊥组成的 L 元向量 v 的集合，其中 V_i 是边 a_i 的值与特殊符号⊥组成的集合，v_i 是 V_i 中元素，特殊符号⊥表示这个位置的边值还没有计算出来。$v^0=<v_1^0, v_2^0, \ldots, v_L^0>$ 表示边值的初始时刻值的向量。

数据流节点是实现计算功能，而边是数据输入和输出。由于计算需要时间，因而边上的数据有了时间的延迟，这样形成了与时间相关的数据流。用符号 v^t 表示 t 时刻的 L 元向量 v，而 v^0 为初始时刻的边向量。

例 2.8[12] 使用数据流图表示 $c=\sqrt{a^2+b^2}$ 的计算过程。

解： 这个计算有 7 个节点：2 个输入、2 个平方、1 个加法、1 个开方和 1 个输出，即节点集合 N={a, b, 平方 1, 平方 2, 加法, 开方, 输出 c}。边的集合 A={(a, 平方 1), (b, 平方 2), (平方 1, 加法), (平方 2, 加法), (加法, 开方), (开方, c)}，这样向量 v 是 6 元向量。

初始时刻是 t0，仅有 a 和 b 的输入数值 3 和 4 是可用的，并作为两个平方节点的输入，因而有 $v^0=<3, 4, ⊥, ⊥, ⊥, ⊥>$。在时刻 t1，两个平方节点计算结果分别为 9 和 16。因此，在时刻 t1 边值向量 $v^1=<3, 4, 9, 16, ⊥, ⊥>$。在时刻 t2，加法节点计算结果为 $v^2=<3, 4, 9, 16, 25, ⊥>$，在时刻 t3，平方根节点计算结果为 $v^3=<3, 4, 9, 16, 25, 5>$，在时刻 t4，所有的边值都计算出来，计算结束，输出结果为 5。图 2-13 是它的数据流图。

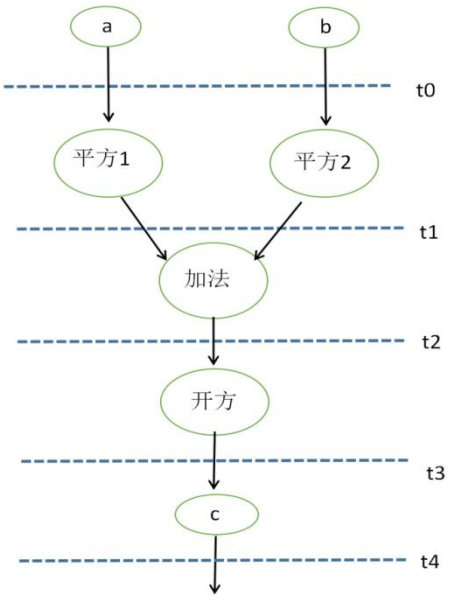


图 2-13 例 2.8 数据流图

2.3.2 数据有限状态机

大多数实时系统具有控制和计算结合的特征。这样必须把有限状态自动机和数据流图结合在一起。一种方法是把时间划分成相等的时间区间，这个时间区间称为状态，而把一个或多个状态布置到数据流图的每个节点。因为，数据流图计算是在数据路上执行的，这样我们称这个组合模型为带有数据流的有限状态机模型 （a finite-state machine with datapath，FSMD），简称为数据有限状态机，或数据有限自动机。

扩展有限状态机 FSM 使得其能处理数据。具体做法是，引入数据变量、数据输入和数据输出。

数据变量集合用 X 表示，其元素使用 x 表示，定义了数据状态，该数据状态定义了在每个节点的所有变量值。因此，使用算术表达式来规范数据变量的值。Expr(X)是数据变量集合 X 上的算术表达式集合，e 或 ei 或 ej 表示其元素，使用巴克斯-诺式 BNF (Backus-Naür Form)方法定义 e 为：

$$e::=k|x|e*e\ (*\in\{+,-,\times\})$$

其中 k 是常值，x∈X 是变量，都是原子算术表达式，而 e1+e2，e1-e2，e1×e2 是复合算术表达式。 使用数据算术表达式定义状态转移条件以及在数据输出情况。

算术逻辑公式 AL（Arithmetic Logic Formula）：

$$AL::=e_1\Delta e_2|AL_1\square AL_2|\neg AL$$

其中 e_1，e_2 ∈Expr(X)是 X 上的算术表达式，Δ∈{≤，<，=，>， ≥}是算术表达式间关系，□∈{∧，∨，→}是逻辑运算符（合取、析取、蕴涵），¬是逻辑运算符非。因而 eiΔej 是算术逻辑原子公式，AL_1□AL_2 与¬AL 是合式公式。如：Data= 0 以及 (a-b)>(x+y)是原子公式，而(counter=0) ∧ (x>10)是合式公式。定义 Data ≠ 0 为¬(Data= 0)，也是合式公式。

再如 cfloor=rfloor 是原子公式，表示电梯所在楼层 cfloor 等于请求的楼层 rfloor，结果电梯保持在原楼层。cfloor ≠rfloor 是合式公式，表示电梯所在楼层 cfloor 不等于请求的楼层 rfloor，结果电梯要么上升要么下降到请求的楼层 rfloor。

定义 2.4 （数据有限状态机 FSMD）FSMD 含有状态集 S，数据变量集 X，输入集 I，输出集 O，转移函数 f 和输出函数 h。

转移函数 f 定义需要状态的转移条件（Transition Condition），为此增加转移条件集 TC，转移条件由算术逻辑公式 AL 定义，如(counter=0) ∧ (x>10)，cfloor ≠rfloor。

把 FSMD 的输入集合 I 分解成数据输入变量集合 I_D 和控制输入变量集合 I_C 组成的二元组，即 I 是 I_D 和 I_C 的笛卡尔积：

$$I=I_D \times I_C$$

这样输入元素是二元组(d，c)，或直接写成 d;c，其中 d 是数据输入变量值；c 是控制输入变量元素。

同样，把 FSMD 的输出集合 O 定义成数据输出变量集 O_D 和控制输出变量集 O_C 的笛卡尔积，

$$O=O_D \times O_C$$

集合 O_D 中元素是赋值语句，如 cfloor:=rfloor，定义了数据输出变量值的改变情况，而 O_C 中元素是控制输出变量向量，如 output<=rfloor-cfloor。这样电梯输出为

cfloor:=rfloor；output<=rfloor-cfloor

其中 cfloor:=rfloor 定义了数据输出：cfloor=rfloor，即电梯要到的楼层是 rfloor 的值， 而 output<=rfloor-cfloor 定义了控制输出：output=rfloor-cfloor，即电梯要上升或下降 rfloor-floor 层：若 rfloor-cfloor>0 则电梯上升 rfloor-floor 层，否则下降 cfloor-rfloor 层，当 rfloor-cfloor=0 时电梯保持不动。

数据有限状态机的转移函数 f 是状态集与输入集的笛卡尔积到状态集的一个映射，刻画了状态转换既依赖于当前状态又依赖于当前的输入值。但带有数据流的有限状态机 FSMD 除了系统的状态外，还有状态上数据变量值。因此把状态集 S 和数据变量集 X 的笛卡尔积 S×X 作为 FSMD 转移函数的基础，其转移实质上是把 S×X 中的元素转移到 S×X 中的元素，这种转移还要依赖于当前数据输入变量的输入值。只有当数据输入变量的当前输入值符合一定条件这种转移才能成功，因此为了使转移函数成功实现转移，还需要增加数据输入变量应该满足的条件，即转移条件。依据这些分析我们给出 FSMD 的转移函数形式化定义：

转移函数 f:(S×X)×I×TC→S×X

同样地分析，给出 Mealy 型输出函数的形式化定义：

Mealy 型输出函数 h:(S×X)×I×TC→O

以两个状态之间转换 FSMD 模型为例，说明转移函数与输出之间的关系。考虑只含有两个状态（si，X）和（sj，Y） 以及两个转移(si，X)→(sj，Y)和(sj，Y)→(si，X)。每个转移的定义都有一个转移条件和输出，输出的动作是可以并行执行。下面的表 2-1 和图 2-14 是两种表达方式：数据输出变量使用 Y:=X+20 形式表示，而控制输出变量使用 output<=1 表示。

表 2-1 数据有限状态机示意表

状态转移	转移条件	数据输出变量；控制输出变量
(si，X)→(sj，Y)	X≤0	Y:=X+20; output<=1
(sj，Y)→(si，X)	Y>0	X:=Y-10; output<=0

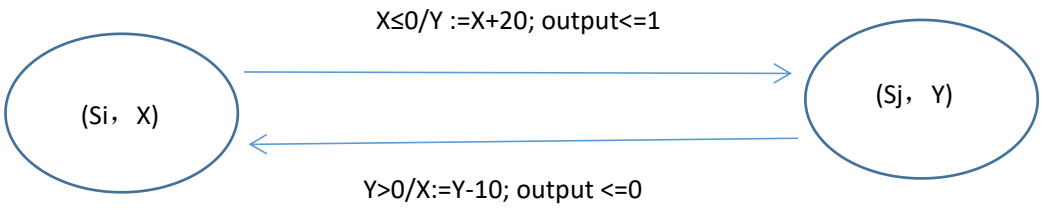


图 2-14 数据有限状态机示意图

由于输入集 I 和输出集 O 都是二元组集合， 即 $I=I_D\times I_C$, $O=O_D\times O_C$，相应地转移函数 f 和输出函数 h 分解成两个函数的积：$f=f_D\times f_C$，$h=h_D\times h_C$，其中

$f_C: S\times I\times TC\rightarrow S$，$f_D: S\times X\times I_D\rightarrow V$,

f_C是状态控制转移函数，定义了下一个状态，这种改变是依赖于当前的状态 s 和当前输入 i=（d，c）和转移条件 AL。转移条件 AL 是依赖于当前数据变量值 d；f_D定义下一个状态的数据变量值，依赖于当前状态 s、当前数据输入变量 x 和数据输入 d，换句话说，对每一个状态 $s_i \in S$，计算每一个变量 $x_j \in X$ 的值，这个值是通过表达式 $e_j \in Expr(X)$来获得的，即 $x_j := e_j$。

输出函数 h 分解成 h_D和 h_C两个函数：

$$h_D: S \times X \times I_D \rightarrow O_D，\ h_C: S \times I_C \times TC \rightarrow O_C$$

其中 h_D定义了数据输出，而 h_C定义了控制输出，相同于 FSM 的控制输出。

FSMD 通常使用 Mealy 型自动机表示，边上权值表现形式为：AL/x:=e;o<=e′，其中 AL 是逻辑公式表示转移条件， x:=e 和 o<=e′都是输出，x:=e 表示数据输出变量 x 的值通过计算算术表达式 e 获得，而控制输出变量输出 o 的值是 e′的值。若使用 Moore 型自动机形式表示，边上权值表现形式为：AL，而状态上表示输出：x:=e;o<=e′。

2.3.3 建模例子

例 2.9 N 层电梯 Mealy 数据有限状态机

设有一座有 N 层高的楼，装有电梯 1 部，每层楼都能到达，建立该电梯的 Mealy 型数据有限状态机 FSMD。

解：定义全局变量 cfloor 存贮电梯的楼层当前状态值：1、2、3、…、N，变量 rfloor 存贮请求要到达的楼层值:1、2、3、…、N。在 FSMD 模型中，可以只使用一个状态 s1，有三个转移都是 s1 到 s1 的。但转移条件和输出（动作）是不同的。

形式化建模：状态集 S={s1}，数据变量集 X:{cfloor，rfloor}，控制输入集 I_C:{}，数据输入变量 I_D:{rfloor}，数据输入集:{1，2，3，…，N}，数据输出变量集 O_D:{cfloor}，控制输出变量集 O_C:{d,u,n}，转移条件集 TC：{cfloor > rfloor, cfloor < rfloor, cfloor=rfloor}。转移函数 f 和输出函数 h 见表 2-2。

表 2-2 例 2.9 转移函数和输出函数表

状态转移	转移条件	数据输出；控制输出
(s1，cfloor)→(s1，cfloor)	cfloor>rfloor	cfloor:=rfloor; d<=cfloor-rfloor
(s1，cfloor)→(s1，cfloor)	cfloor<rfloor	cfloor:=rfloor; u<=rfloor-cfloor
(s1，cfloor)→(s1，cfloor)	cfloor=rfloor	cfloor:=rfloor; n <=0

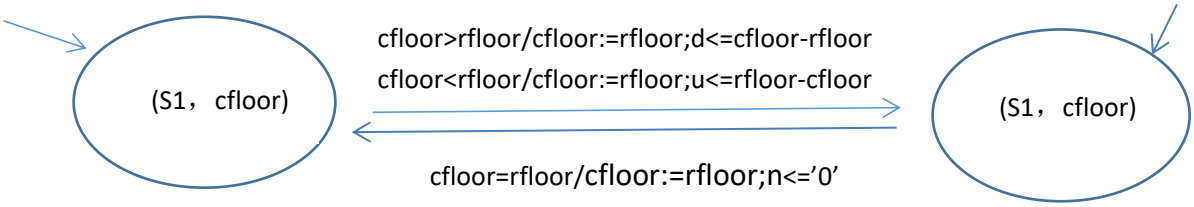


图 2-15 例 2.9 Mealy 型 FSMD

数据输出结果是把 rfloor 的值赋给 cfloor，记住当前电梯所要到达的楼层。控制动作输出结果是： 当 cfloor>rfloor 时，把 cfloor-rfloor 的值赋给控制输出变量 d，电梯下降 cfloor-rfloor 层，到达指定的楼层 rfloor；当 cfloor<rfloor 时，把 rfloor-cfloor 的值赋给控制输出变量 u，电梯上升 rfloor-cfloor 层，到达指定的楼层 rfloor；当 cfloor=rfloor 时，表示请求变量 rfloor 的值就是当前楼层，此时变量 cfloor 保持不变，而控制输出变量 n 的值为 0，表示电梯保持不动状态。

例 2.10 餐巾纸售货机的 FSMD 建模：一款餐巾纸售货机，只接受 5 角和 1 元硬币，1 包餐巾纸价格为 1.5 元。

解：数据输入变量集 I_D:{J，Y}，数据输入变量值集：{0，1}，J=1 表示 J 已接受 5 角，Y=1 表示 Y 已接受 1 元（10 角）。状态集{S0，S5，S10，S15，S20}，其中 S0：表示 0 角，是起始状态，S5：表示 5 角，S10：表示 10 角，S15：表示 15 角，S20：表示 20 角。转移条件集 TC：{J=1，Y=1}。控制输出变量集 O_C：{Open}，控制输出变量值集：{0,1}，数据输出变量集 O_D: {m}，数据输出变量值集：{5}，表示 5 角。在此基础上建立 Moore 型自动机模型和 Mealy 型自动机模型。

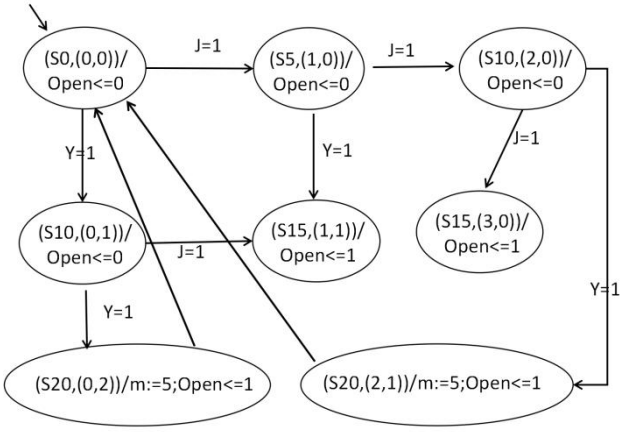


图 2-16 例 2.10 Moore 型自动机 FSMD

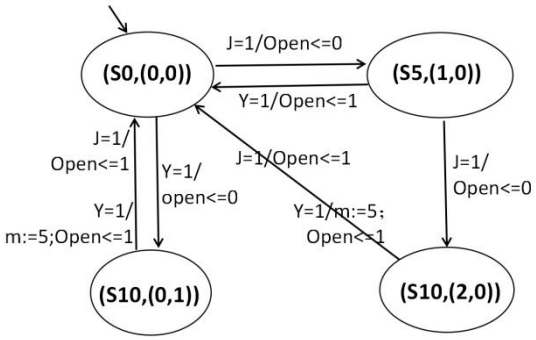


图 2-17 例 2.10 Mealy 型自动机 FSMD

例 2.11 餐巾纸售货机数据有限状态机

一款餐巾纸售货机，可以使用现金 0.5 元和 1 元，也可以使用非现金支付，现金找零。1 包餐巾纸价格为 1.5 元。设计这款餐巾纸售货机数据有限状态机。

解：Moore 型自动机。状态集 S={S0，S1}，其中 S0 是开始状态，表示售货机处于待出货状态，S1 表示售货机售货状态，即打开输出 1 包餐巾纸。数据输入变量集{X}，X 记录支付钱数。转移条件集 TC：{X<15, X>=15}，控制输出变量集 O_C:{Out_C}，控制输出变量值集：{关闭，打开}，数据输出变量集 O_D:{Out_D, X}，数据输出变量取值{0 包，1 包}。

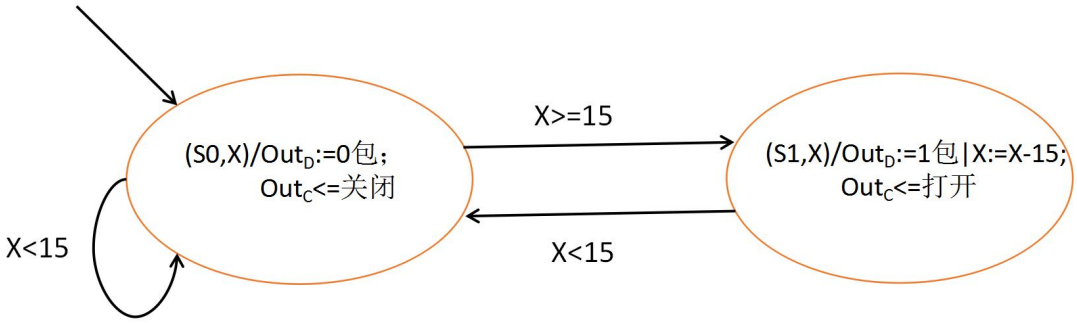


图 2-18 例 2.11 Moore 型 FSMD

Mealy 型自动机模型。状态集 S: {s}，其它相同于 Moore 型自动机，见图 2-19。

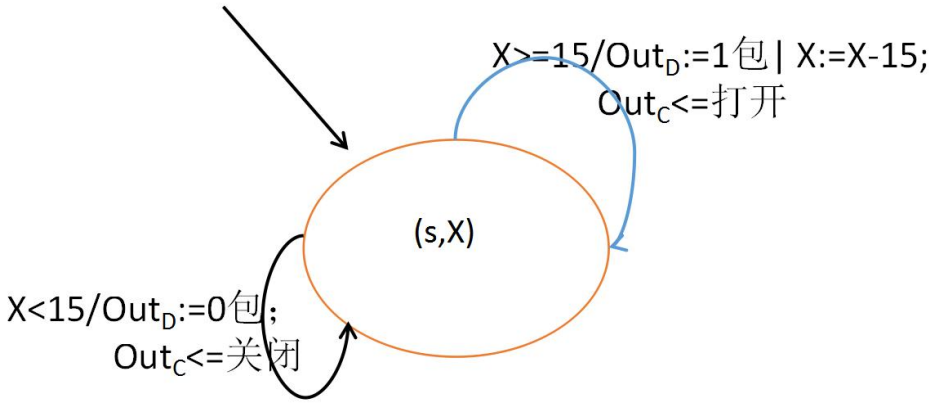


图 2-19 例 2.11 Mealy 型 FSMD

例 2.12 超速自动提醒系统

为了安全驾驶，在车子启动后启动车速提醒系统并设置提醒车速值 100 公里/小时，同时显示车速。当车速超过这个提醒值 100 公里/小时时，汽车会自动语音提醒“超速”，直到车速低于提醒值 100 公里/小时为止。

解：形式化建模：状态集 S={不提醒，提醒}，数据输入变量集 X:{Carv}，其中 Carv 表示车速，控制输入变量集 I_C:{}，数据输入变量值集 I_D:[10，120]为 Carv 取值范围，数据输出变量集 O_D:{Outputspeed}，控制输出变量集 O_C:{Outputsound}，控制输出变量值集：{超速}，转移条件集 TC：{Carv >=100 公里/小时, Carv <100 公里/

小时}。转移函数 f 和输出函数 h 见图 2-20。

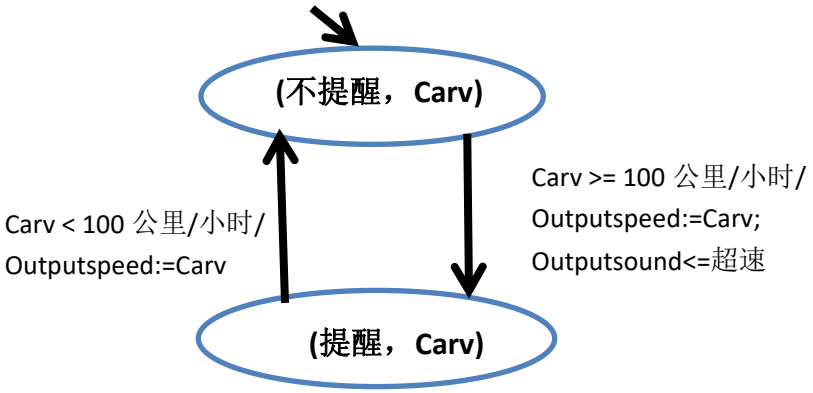


图 2-20 例 2.12Mealy 型自动机 FSMD

例 2.13 加热空调

加热空调是一个智能空调， 能依据房间的温度自动地开和关。譬如设置房间温度为 20°C，空调控制系统会围绕 20°C 进行控制：当室温大于等于 22°C 时空调关闭，房间自动降温，当房间温度低于 18°C 时，空调自动打开，对房间进行加温，这样反复进行。

解： 形式化建模。空调状态集 S={关，开}，初始状态为关，数据输入变量集 X={T}，其中 T 是房间温度，数据输入变量值集 I_D: [0, 30]为 T 的取值范围，数据输出变量集 O_D: {Temp}（显示房间温度），控制输出变量集 O_C: {Output}，控制输出变量值集：{开，关}，转移条件集 TC={T≥22°C，T≤18°C}。状态转移函数 f 和输出函数 h 的定义如图 2-21。

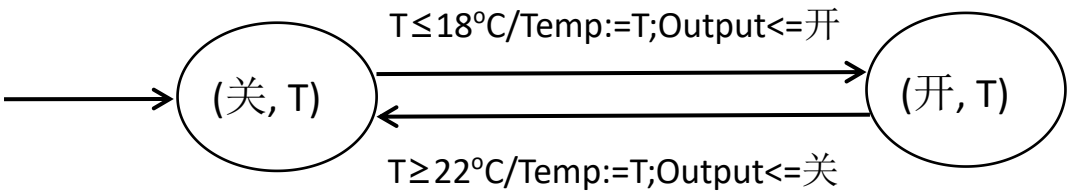


图 2-21 例 2.13 Mealy 型自动机 FSMD

第 2.4 节 混成自动机 HA（Hybrid Automata）

混成系统既有离散状态也有连续状态。混成自动机是描述混成系统的一种建模方法和技术，从离散和连续两个方面描述系统的变迁和演化。本节内容可参阅文献[4，13]。

2.4.1 混成系统

混成系统是一个动态系统，刻画了连续（实值）状态和离散（有限值）状态之间的交互，反映了状态随时间的演化。动态系统可以通过外在输入而被激活，这些外在输入可能是控制信号（如：驾驶员发给飞行器的起飞命令、驾驶员给汽车的制动命令或自主巡航命令或自动驾驶命令等），也可能是不可控制的干扰（如：影响飞行器的风，影响汽车行驶的障碍物、影响汽车刹车制动的路面等）。一些动态系统也可能需要有输出，这些输出可能是被测量的值（如：飞行器高度和速度、汽车的速度等），也可以是展示系统的状态（如：飞行器状态正常、汽车发动机正常等）。带有输入和输出的动态系统称为控制系统。

动态系统可以分成三类：

连续型： 状态在 n 维欧式空间 R^n（n>=1）取值，用 $x \in R^n$ 表示连续动态系统的状态，是 n 元向量。例如：房间温度（1 元向量）、汽车速度和加速度（2 元向量）、飞机速度-仰角-空间位置（3 元向量）等。

离散型：状态在有一个有限集或一个可数集{q1，q2，…}中取值，用 q 表示离散系统的状态。如：一个灯光开关 q∈{ON， OFF}，高铁运行状态∈{加速、减速、匀速}。

混成型： 系统既包含连续状态又含有离散状态，因此混成型系统的一部分状态在欧式空间 R^n 中取值，而另一部分状态在一个有限集取值。如：智能空调器、汽车自动换档器、高铁运行系统等智能系统。

一个智能空调系统是保持房间在一个指定温度的自动控制系统，这个系统由开关自动控制，无论在空调处于开或关状态时，房间温度变化都是服从某个微分方程，比如：dx / dt=-a(x-30)，dx / dt=-bx,其中 x 是房间温度变量， t 是时间变量，a 和 b 是系数，分别反映空调制热能力系数和房间保温系数。通过这些微分方程，可以求得温度与时间的显示函数关系：$x=k_a e^{-at}+30$, $x=k_b e^{-bt}$。

2.4.2 混成自动机

使用混成自动机来建模混成系统，其节点由连续状态和离散状态组成，代表连续状态的变化，而边代表离散状态的转移。

定义 2.5（混成自动机 Hybrid Automaton, HA）一个混成自动机 H=(Q, X, F, Init, D, E, G, R)，由 8 部分组成，(q, x)∈Q×X 是 H 的状态，

1、Q={q1, q2, . . . }是离散状态集；

2、$X=R^n$是连续状态集，P(X)是 X 的幂集，其元素是由若干连续变量应满足的条件组成，这里条件可以是线性方程，也可以是微分方程；

3、F(•, •): Q×X→R^n是向量场函数，为离散状态指定一组连续变量变化应服从的方程；

4、Init⊆ Q×X 是初始状态集；

5、Dom(•): Q →P(X)是域函数，为每个离散状态指定连续变量停留在这个离散状态应满足条件（一般是线性条件），即定义了离散状态的连续变量域；

6、E⊆Q×Q 是边集，刻画离散状态的转换；

7、G(•): E →P(X)是转换条件，为每一个边指定一组连续变量满足的条件，它激活离散状态的转换；

8、R(•, •): E×X→ P(X)是重置映射， 为边上的连续变量向量赋值一组函数方程，重新赋值这些连续变量。

例 2.14 制热空调系统

一个制热空调系统是保持房间在一个 20°C 的系统，空调由开关自动控制，当室内温度低于 19 度时空调开始启动‘ON’状态并对房间进行加温，此时室内温度变化服从微分方程：dx / dt=-a(x-30)；当室内温度上升到 21 度时，空调启动‘OFF’状态，室内温度变化服从微分方程:dx / dt=-bx。参数 a 反映了空调制热能力， 而参数 b 反映了房间保温能力。考虑到一些外在不确定因素，如温度检测动态性，室内温度可能已经超出了范围，因此再规定‘ON’状态和‘OFF’状态内部约束条件，如：在‘OFF’状态时室内温度大于等于 18 度， 而在‘ON’状态时室内温度小于等于 22 度。

解： 建立这个空调系统的混成自动机模型：

1、离散状态集 Q={ON，OFF}；

2、连续状态集 X=R，连续变量 x 代表房间的温度，它是时间 t 的函数；

3、向量场函数 F(・,・):{ON，OFF}×X→R：

F(ON，x)=(dx／dt=-a(x-30))，空调启动室内温度上升，

F(OFF，x)=(dx／dt=-bx)，空调关闭房间温度自行下降;

4、初始状态集 Init：{OFF}×{x∈R|x<=15}；

5、域函数 Dom(・)：Q →P(X)定义为：

Dom(ON)={x<=22}，规定 ON 状态下室内温度不超过摄氏 22 度，

Dom(OFF)={x>=18}，规定 OFF 状态下室内温度不低于摄氏 18 度；

6、边集 E⊆Q×Q：

ON→OFF：ON 状态到 OFF 状态有条边，

OFF→ON：OFF 状态到 ON 状态也有一条边；

7、转换条件 G(・)：E →P(X)：

G(ON→OFF) ={x>=21}，从 ON 状态转换到 OFF 状态的条件是室内温度大于等于摄氏 21 度，

G(OFF→ON)={x<=19}，从 OFF 状态转换到 ON 状态的条件是室内温度小于等于摄氏 19 度；

8、重置映射 R(・,・)：E×X→ P(X)：

为每个边都指定了一个空集，即没有重置动作，在状态转换过程中室内温度变量 x 不进行重置，保留变换前的值。

2.4.3 混成自动机图形化

同有限状态机一样，使用有向图表示混成自动机。

给了一个混成自动机 H=(Q，X，F，Init，D，E，G，R)，把 Q 中元素（状态）作为定向图的节点，把 E 作为有向图的有向边，

1、对每一个节点 q∈Q，都关联一个连续状态集{x∈X|(q，x)∈Init}，一个向量场 f(q，・)： $R^n \to R^n$（连续状态 x 应该服从的微分方程组）和一个域 $Dom(q) \subseteq R^n$ 连续状态停留在这个节点的条件；

2、对每一个起始于状态 q∈Q 终止于 q’∈Q 的有向边 q→q’，都关联一个转换条件 $G(q \to q') \subseteq R^n$，规定了离散状态 q 转换到离散状态 q’的转移条件集，以及一个重置函数 R(q→q’，x)：$R^n \to P(R^n)$，重置语句用新值对连续状态进行重置。

例如：例 2.14 空调系统 HA 图形化：

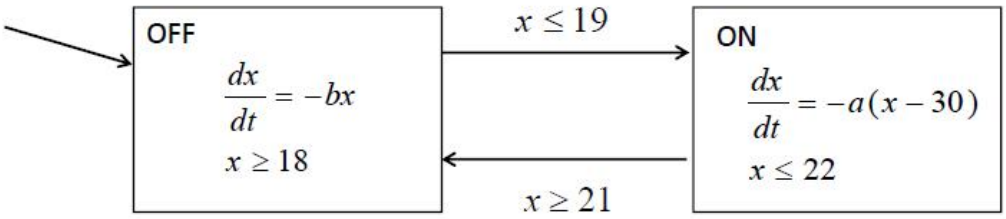


图 2-22 例 2.14 加热空调器 HA 图形化

注： 例 2.14 空调系统中，向量场函数 F(•,•)规定了空调温度连续变化应该服从微分方程。在空调启动状态时，温度 x 按照 $x=k_a e^{-at}+30$ 函数往摄氏 30 度上升，而在关闭状态时，房间温度 x 按照 $x=k_b e^{-bt}$ 函数往摄氏 0 度下降。

现在来分析一下，参数 k 和参数 a 与 b 对系统的影响。现在假设室温为摄氏 15 度，则空调启动进行加热，将 x=15 代入公式 $x=k_a e^{-at}+30$ 求得 $k_a=-15$，因为时间 t=0。假定时间 t 后房间温度上升到摄氏 20 度，则反映空调能力的参数 a=-1/t*ln(2/3)，明显参数 a 的取值与时间有关。若 10 个单位时间房间温度上升到了摄氏 20 度则得到 a=0.04。若在假定 5 个时间单位后房间温度上升到 20 度，则 a=0.08。现在考虑房间自行保温参数。若开始时房间温度为摄氏 21 度则时刻 t=0，从而有参数 $k_b=21$。设过时间 t 房间温度下降到 19 度，则有 $19=21e^{-bt}$，求得 t=0.1/b。若要求房间温度从 21 度下降到 19 度需要 5 个时间单位分钟， 则得到参数 b=0.02。我们可以得到一组参数值 $(k_b, b)=(21, 0.02)$。

因此，当常数 a 和 b 取定值后，如 a=0.08,b=0.02，这空调器的效能以及房间保温能力就确定了，其图形化为图 2-23。

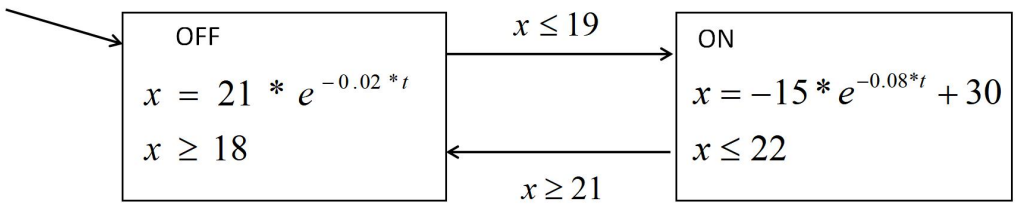


图 2-23 例 2.14 加热空调器 HA 图形化

2.4.4 混成系统建模例子

例 2.15 （**水缸系统**）由两个水缸组成的水缸自动控制系统，两个水缸中的水都

是按照常速流出，水是按照一个常速通过一个软管流进水缸，在任何时刻水只能流进其中一个水缸，假定软管在两个水缸瞬间转换，并保持两个水缸中的水在一定容量之上。

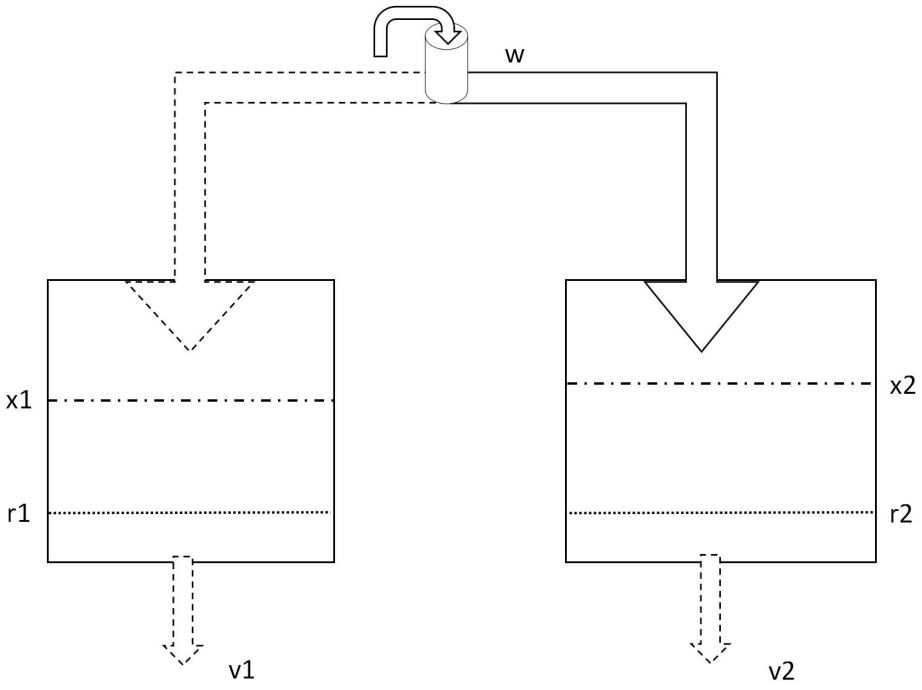


图 2-24 例 2.15 水缸示意图

分析：设 x1，x2 分别表示第 1 水缸和第 2 水缸中水的容量，v1 和 v2 分别表示水缸 1 和水缸 2 的出水(常)速度。设 w 表示水流进系统的(常)速度，因此 w-v1 和 w-v2 分别是第 1 水缸和第 2 水缸净(实际)进水速度，都是常速度。往第 2 水缸注水时，保持第 1 水缸水容量大于等于 r1 和第 2 水缸水容量大于等于 r2。同样，往第 1 水缸注水时，保持第 2 水缸水容量大于等于 r2 和第 1 水缸水容量大于等于 r1。为了实现这个需求，需要一个控制系统自动改变调整软管往水缸 1 注水（当水缸 1 的容量 x1≤r1 时），以及调整软管往水缸 2 注水（当水缸 2 的容量 x2≤r2 时）。

解：构建混成自动机模型

1、离散状态集 Q={q1，q2}，q1 是第 1 水缸，q2 是第 2 水缸。

2、连续状态集 $X=R^2$， 连续变量 x 是二维向量(x1，x2)，x1 代表第 1 水缸水容量，x2 是第 2 水缸水容量，都是时间 t 的函数。

3、向量场函数 F(•,•)：{q1，q2}×X→R^2：

F(q1，x)=(dx1 / dt=w−v1，dx2/dt=−v2)，w−v1 是第 1 水缸的净进水速度，−v2 是第 2 水缸出水速度；

F(q2，x)=(dx1 / dt=−v1，dx2/dt=w−v2)，−v1 是第 1 水缸出水速度，w−v2 是第 2 水缸净进水速度。

4、初始状态集 Init：{q1，q2}×{x∈R^2|x1>=r1∧x2>=r2}：

系统开始时第 1 水缸和第 2 水缸容量分别大于等于 r1 和 r2。

5、域函数 Dom(·)：Q →P(X)定义为：

Dom(q1)={x∈R^2|x2>=r2}，当第 1 水缸进水时保持第 2 水缸的容量不低于 r2；

Dom(q2)={x∈R^2|x1>=r1}，当第 2 水缸进水时保持第 1 水缸的容量不低于 r1。

6、边集 E⊆Q×Q：{q1→q2，q2→q1}，转换是在两个水缸间进行。

7、转换条件 G(·)：E →P(X)：

G(q1→q2) ={x∈R^2|x2<=r2}，在第 1 水缸进水时，只要第 2 水缸的容量小于等于 r2，自动机状态就转化到第 2 状态，即往第 2 水缸注水；

G(q2→q1) ={x∈R^2|x1<=r1}，在第 2 水缸进水时，只要第 1 水缸的容量小于等于 r1，自动机状态就转化到第 1 状态，即往第 1 水缸注水。

8、重置映射 R(·,·): E×X→ P(X)：R(q1→q2，x)=R(q2→q1，x)={x:=x}，状态改变时连续状态不改变，即保持水缸的当前容量。

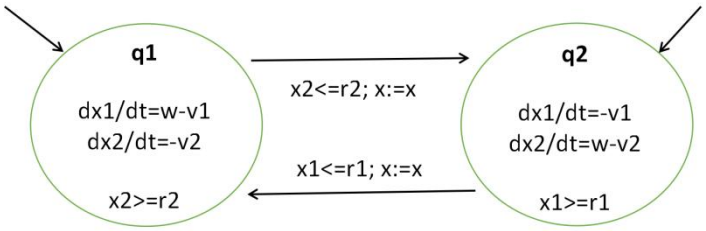


图 2-25 例 2.15 水缸系统混成自动机模型

2.4.5 混成自动机演化

混成自动机刻画了动态系统随着时间变化的演化，下面考虑混成自动机状态(q(t)，x(t))的可能演化：

- 从初始状态(q0，x0)∈Init 出发，连续状态 x 服从微分方程 dx/dt = f_{q0}(q0，x(t))以及初始 x(0) = x0，离散状态 q(t)保持初始离散状态 q0。

- 连续演化重复进行，只要 x(t)保持在域 Dom(q)里：
  - 假设混成自动机进入到了离散状态 q，时间到了 t 时刻，则连续变量 x 的值为 x(t)，并服从于微分方程 $dx/dt = f_q(q(t)，x(t))$而进行演化，离散状态 q(t)保持常状态 q，即 q(t)=q；
  - 如果在时刻 t 的后续某个时间点 t’，连续状态 x 满足某个边 (q，q’) ∈ E 的转换条件 $G(q\rightarrow q') \in R^n$，则离散状态从 q 转移到离散状态 q’；连续状态 x(t)从重置 $R(q\rightarrow q'，x(t))\in R^n$ 中获得新值，并在离散状态 q’状态下服从于微分方程 $dx/dt = f_{q'}(q'，x(t))$而演化；
  - 连续变量 x(t)随着时间进行演化，触发离散状态的转换，在离散状态转换后，连续演化重新开始，整个过程重复进行。

- 对所有离散状态 q ∈ Q，域函数 Dom(q)中的函数都是 Lipschitz 型连续函数。
- 最后，假定对于所有的边 e∈E，边的转换条件 G(e)≠Ø（空集），以及所有的连续状态 x∈G(e)， 连续变量 x 的重置操作 R(e，x)≠Ø。

注：在数学中，Lipschitz 型连续函数是指满足 Lipschitz 连续条件的实值函数。利普希茨连续条件（Lipschitz continuity），以德国数学家鲁道夫·利普希茨命名，是一个比通常连续更强的条件。对于在实数集 R 的子集 D 上函数 f：D→R，若存在非负常数 k，使得|f(a)-f(b)|≤k|a-b|，则称 f 满足利普希茨条件，对于最小常数 k 称为利普希茨常数。绝对值函数 f(x)=|x|是 Lipschitz 型连续函数，但不是可微函数。

为了刻画混成自动机 H=(Q，X，F，Init，D，E，G，R)的具体演化， 把时间集分化成连续区间，使得在连续区间上很好地体现连续状态的演化，同时又能区分离散状态的转换点。这样的时间区间集称为混成时间集。

定义 2.6 (混成时间集) 一个混成时间集是一列区间 $T=\{I_0，I_1，I_2，...，I_N\}=\{I_i\}_{i=0}^{N}$ 是有限集或者无穷集（N=∞）使得对于所有的 i 都有 $I_i=[t_i, t_i'，]$并且 $t_i\leqslant t_i'=t_{i+1}$；若 N<=∞则或者 $I_N=[t_N，t_N']$或者 $I_N=[t_N，t_N')$。

时间点 t_i'是在离散迁移前一时刻，而 t_{i+1}是离散迁移后的那一时刻。为了时间点具有连续性，规定时间区间 I_i的右端点 t_i'和时间区间 I_{i+1}的左端点 t_{i+1}重合。因此，这个时间点恰好是混成自动机离散状态转换发生时间点。这样我们假定离散迁移是瞬时发生的。注意到 $t_i=t_i'$，即区间 I_i是单点集$\{t_i\}$，这种情况也是可能

发生的。

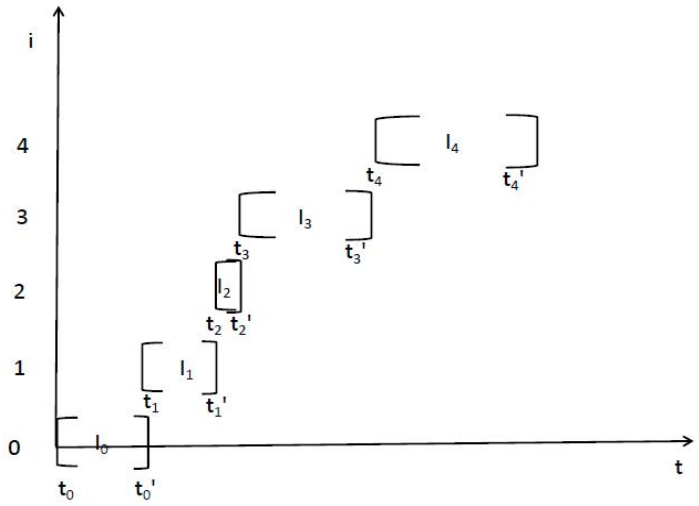


图 2-26 混成时间区间示意图

例 2.16 制热空调系统

以图 2-23(b)制热空调系统为例，建立空调系统混成自动机演化过程。取空调加热方程为 $x=-15e^{-0.08t}+30$，房间冷却方程为 $x=21e^{-0.02t}$，房间初始温度为 15 度。

解：开始时间区间 I_0：房间加热状态，t=0，x(0)=15，房间初始温度为摄氏 15 度， 经过 6.4 个单位时间加热后，房间温度上升到摄氏 21 度，即 x(6.4)=21，空调进入冷却状态，I_0=[0，6.4]；

第二个时间区间 I_1：房间冷却状态，房间从摄氏 21 度冷却下降到摄氏 19 度需要 5 个单位时间，x(11.4)=19，空调进入到加热状态，得 I_1=[6.4，11.4]；

第三个时间区间 I_2： 房间加热状态，此时加热方程参数 k=11，空调服从温度方程 $x=-11e^{-0.08t}+30$，从摄氏 19 度加热到摄氏 21 度，需要 2.5 个单位时间，得 I_2=[11.4，13.9],空调转移到冷却状态；

第四个时间区间 I_3：房间冷却状态，冷却的初始温度为摄氏 21 度，下降到摄氏 19 度为止，需要 5 个单位时间，得 I_3=[13.9，18.9],空调转移到加热状态；

第五个时间区间 I_4：房间加热状态，空调服从温度方程 $x=-11e^{-0.08t}+30$，从摄氏 19 度加热到摄氏 21 度，需要 2.5 个单位时间，得 I_4=[18.9，21.4],空调转移到冷却状态；

这样一直重复下去，直到空调关机为止。

I_0=[0，6.4]，I_1=[6.4，11.4]，I_2=[11.4，13.9]，I_3=[13.9，18.9]，I_4=[18.9，21.4]。

空调温度控制演化过程示意图如图 2-27。

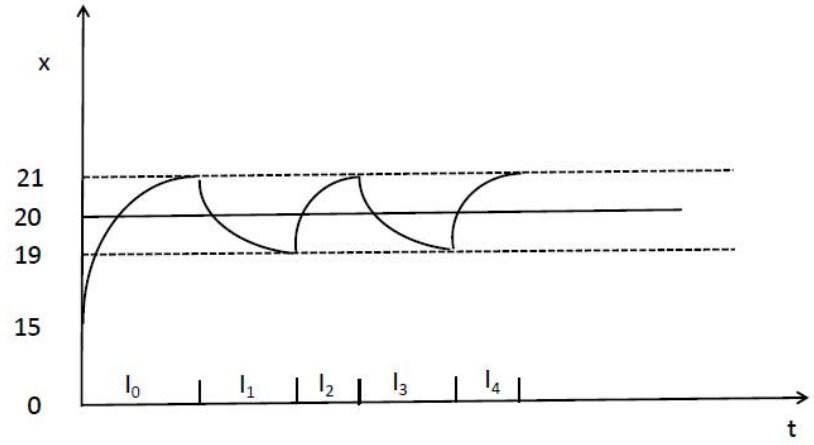


图 2-27 例 2.16 空调温度控制演化过程示意图

例 2.17 水缸系统

设第 1 水缸和第 2 水缸中水的容量都是 1，水缸 1 和水缸 2 的水流(常)速度为 1/2，即 v1=v2=1/2，水流进水缸也为常速度为 3/4，即 w=3/4，因此，第 1 水缸和第 2 水缸进水速度，都是常速度 3/4-1/2=1/4，再设 r1=r2=0。

解： 初始状态是 q=q1，x1=0，x2=1，即第 1 水缸无水，第 2 水缸满缸，因此初始状态往第 1 水缸注水。第 2 水缸是满缸(x2=1)，出水速度 v2=1/2，水缸流干需要 2 个单位时间，第 2 水缸流干后系统自动往第 2 水缸注水，此时第 1 水缸水容量 x1=1/4*2=1/2，即半缸水，时间区间 I_0=[0，2]。

系统状态为 q=q2，x1=1/2，x2=0。系统往第 2 水缸注水，第 1 水缸出水速度 v1=1/2，而 x1=1/2，因此经过 1 个单位时间后，第 1 水缸水流干，系统转向往第 1 水缸注水，此时第 2 水缸水容量 x2=1*1/4=1/4，时间区间 I_1=[2，3]。

系统状态为 q=q1，x1=0，x2=1/4，系统往第 1 水缸注水。由于此时第 2 水缸只有 1/4 水容量，出水速度 v2=1/2，因此经过 0.5 个单位时间后，第 2 水缸无水，系统转向往第 2 水缸注水，此时第 1 水缸水容量 x1=1/4*0.5=1/8，时间区间 I_2=[3，3.5]。

系统状态为 q=q2，x1=1/8，x2=0，系统往第 2 水缸注水。由于第 1 水缸只有

1/8 水容量，出水速度 v1=1/2，因为经过 0.25 单位时间，第 1 水缸无水，系统转向往第 1 水缸注水，此时第 2 水缸水容量 x2=1/4*0.25=1/16，时间区间 I_4=[3.5，3.75]。

系统状态为 q=q1，x1=0，x2=1/16，系统往第 1 水缸注水。第 2 水缸水容量 x2=1/16，出水速度 v2=1/2，因此经过 0.125 单位时间，第 2 水缸无水，系统转向第 2 水缸注水，此时第 1 水缸水容量 x1=1/4*0.125=1/32，时间区间 I4=[3.75，3.825]。

系统状态为 q=q2，x1=1/32，x2=0，系统往第 2 水缸注水。第 1 水缸经过 1/16 单位时间后无水，系统转向第 1 水缸注水，此时第 2 水缸水容量 x2=1/4*1/16=1/64，时间区间 I_5=[3.825，3.888]。

得到混成时间集 I={[0，2]，[2，3]，[3，3.5]，[3.5，3.75]，[3.75，3.825]，[3.825，3.888]}，两个水缸水容量演化过程如图 2-28。

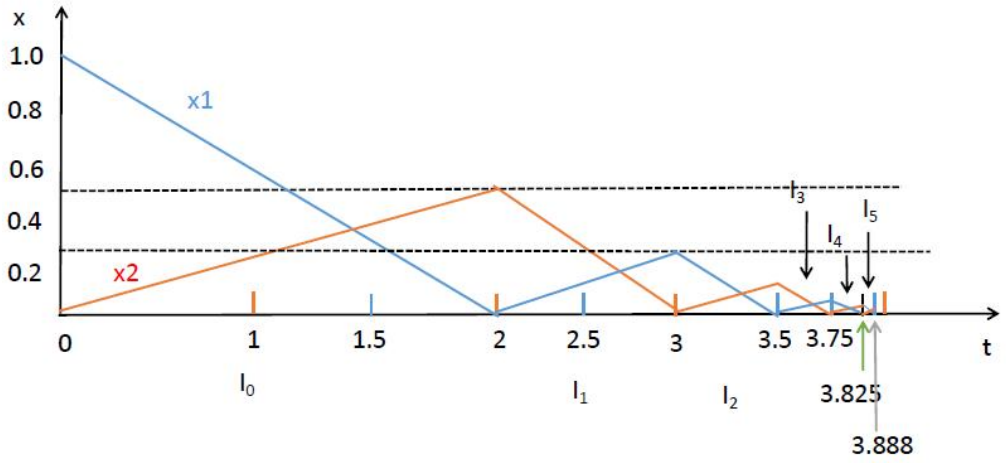


图 2-28 例 2.17 水缸水容量演化过程图

第 2.5 节 图形化建模语言 SysML

本节将介绍图形建模语言 SysML(System Model Language)。 SysML 是面向系统工程的一种图形建模语言，可以用来可视化设计各种规模的工程技术系统，通常--由硬件、软件、数据、人和过程组成的系统。系统工程师会负责对工程技术系统进行规范、分析、验证和检验。SysML 建模工具有 Modelio(创建者：Modeliosoft) 、Papyrus(创建者： Atos Origin)。文献[14]是一个简单明了的 SysML 入门教程。本段的例子大多数来自文献[15，16]。

2.5.1 SysML 介绍

SysML 是系统工程领域的标准化建模语言，提供了创建系统模型时使用的图形语言，包含了能够表示特殊意义的图形标识。SysML 可以用来创建系统结构、行为、需求和约束的图形模型。

SysML 由九种类型图组成，分别是包图、需求图、活动图、序列图、状态机图、用例图、参数图、块定义图、内部块图，其中活动图、序列图、状态机图、用例图统称为行为图， 而块定义图和内部块图统称为结构图。

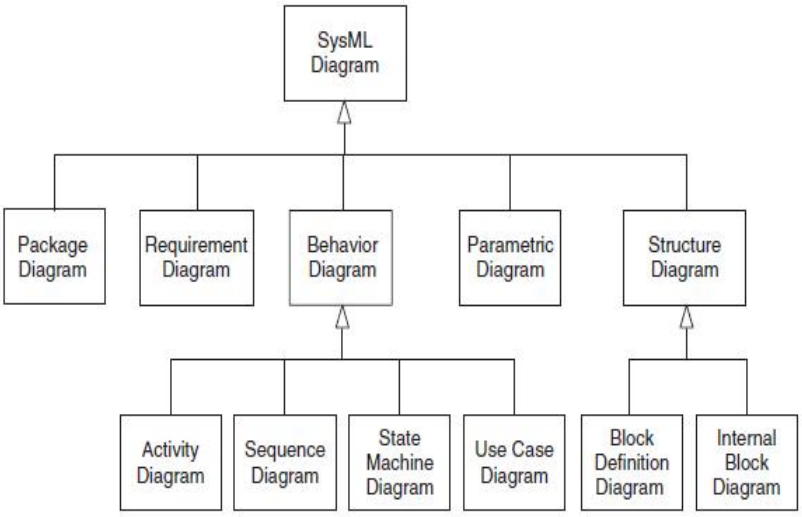


图 2-29 SysML 图分类

SysML 的通用图框架如图 2-30 所示，每个 SysML 图都由图外框、头部以及内容区域三部分组成。

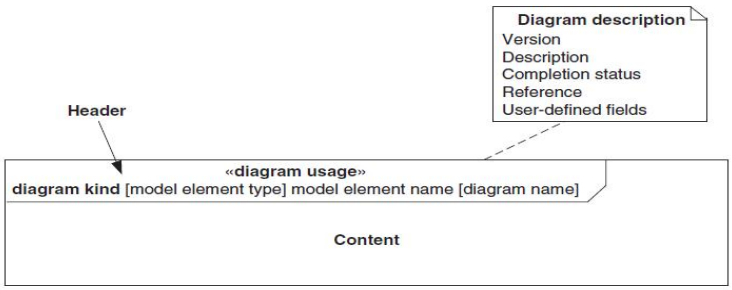


图 2-30 SysML 通用框架图[15]

图外框是指图的外部黑色实线，在 SysML 中外框不能省略。内容区域（Content）是存放 SysML 模型元素的地方。头部（Header）位于图的左上角，对模型图的类型、名称、模型元素类型及名称进行概要性描述。若要提供关于图表状态和用途的更多详细信息可选图表说明（Diagram description），可附加到框架边界。

头部（Header）模型图描述格式是固定的：

diagram kind [model element type] model element name [diagram name]，

其中 diagram kind 指的是图类型，model element type 是模型元素类型，model element name 是模型元素名称，diagram name 是图名称。

图类型的命名只能在 SysML 定义的图类型缩写集合中选择，用户不能随意命名。

表 2-3 图缩写表

图类型	对应图缩写
包图	pkg
需求图	req
活动图	act
序列图	sd
状态机图	stm
用例图	uc
参数图	par
模块定义图	bdd
内部模块图	ibd

SysML 定义了模型元素类型集合（model element type），这些类型的模型元素在图中不能任意选择，每种 SysML 图中所能表达的模型元素是有规则限制的.

模型元素名称(model element name): 用户自定义的模型元素的名称。

图名称(diagram name): 用户自定义的图的名称。

表 2-4 可表达模型元素类型表

图类型	可表达的模型元素类型
包图	package、model、modelLibrary、view
需求图	package、model、modelLibrary、view、requirement
活动图	activity
序列图	interaction
状态机图	stateMachine
用例图	package、model、modelLibrary、view
参数图	block、constraintBlock
模块定义图	package、model、modelLibrary、view、block、constraintBlock
内部模块图	block

2.5.2 SysML 建模工具 EA

EA(Enterprise Architect) 是用于软件系统的设计与开发、企业业务过程建模以及更广泛建模的可视化平台。Enterprise Architect 基于 UML 2.3 规范，是一款不断进步和完善的工具，它覆盖了开发周期的所有方面，提供了从初始设计阶段到系统部署、维护、测试以及修改控制的全程可跟踪性。EA 的功能分类罗列如下。

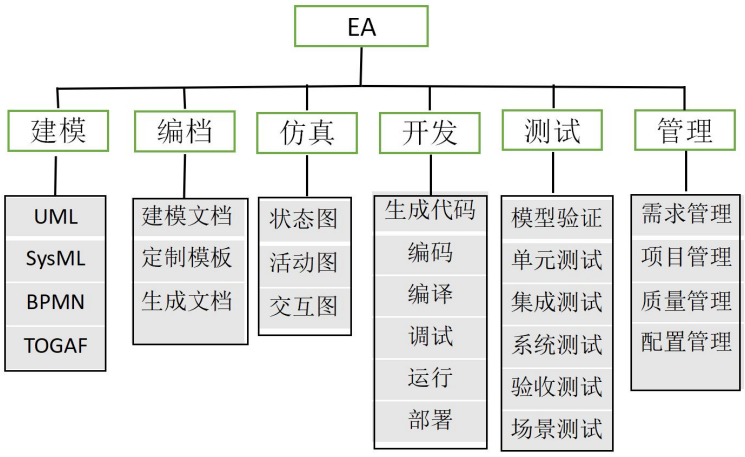


图 2-31 EA 功能分类

EA 具有如下几个优点：

- 用于系统和软件应用程序的可视化模型驱动开发。
- 通过原型快速设计，及早纠正错误，降低成本。
- 自动实施一致性检查，提升敏捷性，并通过协作提高重用性，降低经常性和

非经常性费用。

- 与扩展的设计团队共享、协作和审查由 Rational Rhapsody 或其他设计工具生成的工程生命周期工件。

2.5.3 SysML 建模介绍

本段介绍 SysML 中九种图的建模方法。文献[15]以机动车建模等为例全面介绍如何使用 SysML 进行建模，文献[16]以载人航天活动的最后阶段——控制降落伞开启的降落伞系统[17]为例，介绍如何使用 SysML 进行建模，并进行降落伞系统的一致性仿真验证。 本段将选这两个文献中的 SysML 图进行介绍。

2.5.3.1 包图

包图是显示系统模型的组织方式时所创建的图。在包图中显示各种类型的元素和关系， 以表达系统模型结构。

在项目开始时，第一次创建模型结构时，就会创建多个新的包以及新的包图。当需要修改模型结构的时候，就会创建新的包图。

包图的图类型缩写是 pkg。

包图的图形式为：pkg [model element type] package name [diagram name]

图类型是 pkg，模型元素类型（the model element type）可以是 model，package，model library，或 view。

图 2-32 是自动驾驶领域的 SysML 包图，从包图中可以看到，自动驾驶领域的 SysML 建模共用了包括了用例图、需求图、参数图在内 10 种图。

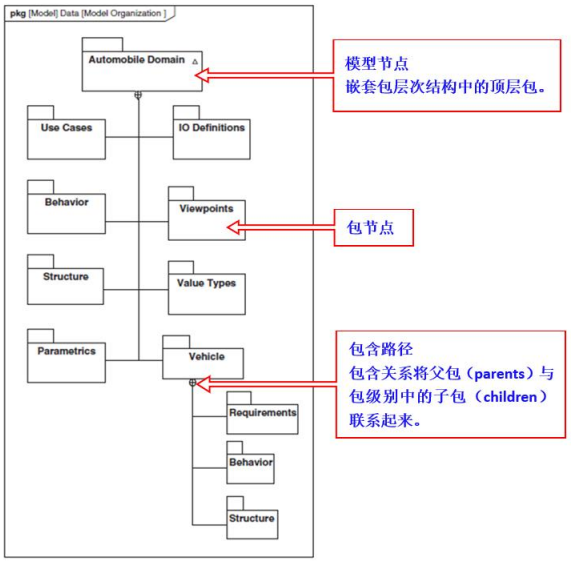


图 2-32 自动驾驶包图[16]

2.5.3.2 需求图

需求图是以图形式表达系统需求的各方面信息，它是传统基于文字需求的 SysML 图形表示。

需求图的标识法是一个矩形，在名称之前有元类型<<requirement>>。需求图的图形类型是 req。

载人航天活动返回舱降落伞系统的主要任务是及时有效地打开降落伞进行减速，保证返回舱按照规定的安全速度着陆。主需求要求返回舱降落伞系统能够在正常状态下开伞成功并且能处理遇到的故障，同时该需求进一步分解为 2 个子需求，一是主降落伞系统完成开伞任务，另一个是在主降落伞系统失效时，备用降落伞系统执行开伞功能。如图 2-33。

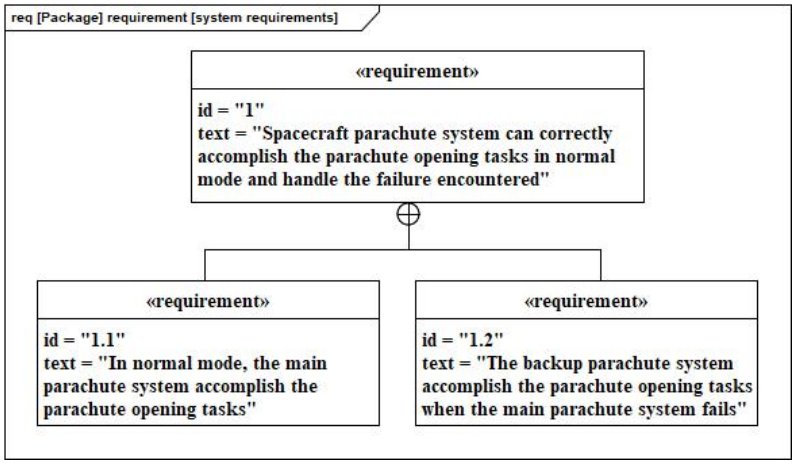


图 2-33 载人航天返回舱降落伞系统需求图[16]

返回舱降落伞系统的高层需求被精化为多个子需求。需求 1.1 派生出了 3 个子需求，其中子需求 2.1 要求减速伞能够将返回舱的速度降到亚声速。除此之外，需求的分配关系也显示在图中，子需求 2.1 与 2.2 被分配到主降落伞系统，而子需求 2.3 被分配到伞舱盖弹射分离系统。如图 2-34。

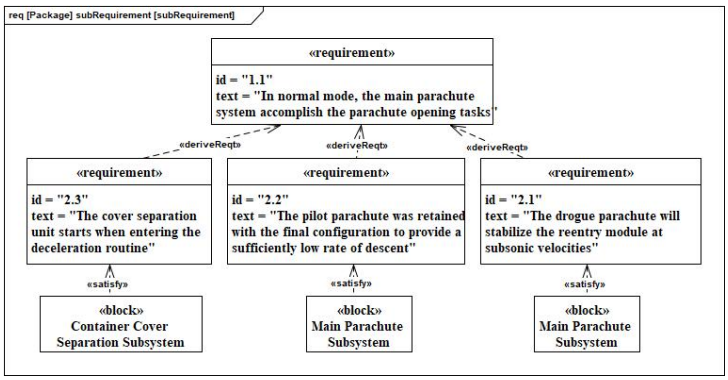


图 2-34 返回舱降落伞系统需求精化图[16]

2.5.3.3 活动图

活动图是能够表达系统动态行为信息的三种 SysML 图（活动图、序列图、状态图）之一，是唯一能够说明连续行为的图。它可以表达各种各样的活动以及复杂的控制逻辑。

活动图是一种行为图，它是系统的一种动态视图，显示随着时间的推移行为和事件的发生序列。活动图通过行为表示对象-事件、能量、或者数据的流动，关注系统在操作时，系统对象是如何在行为执行过程中被访问和被修改的。在表达系统及执行者期望的行为时，需要创建系统的行为图。活动图可以很好地处理输入和输出复杂控制流。

返回舱降落伞系统的外部环境实体雷达负责发送高度信息给返回舱降落伞系统，它每隔 2 秒向外发送高度信息。图 2-35 描述了雷达工作流程。

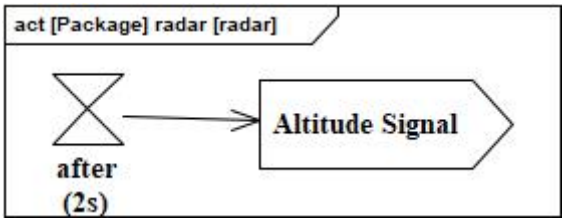


图 2-35 雷达信息工作活动图[16]

返回舱降落伞系统正常工作流程如下：当返回舱距离地面高度为 10 千米左右，主降落伞系统开始工作，先后拉出引导伞、减速伞和主伞，使返回舱的速度缓缓下降。主降落伞系统工作时会接收来自伞舱盖弹射控制系统以及状态监测系统发出的信号。伞舱盖弹射控制系统完成弹出伞舱盖的任务。状态监测系统监测主降落伞工作状态并正确判断系统故障从而切换至备用降落伞系统。为了保障人员的安全，空中救生系统监测返回舱状态并及时开启故障应急救生系统。

主伞状态监测开启主伞调用行为具体流程如下：主降落伞系统接收 GNC（GNC 系统：飞船制导(Guidance)、导航(Navigation）与控制（Control）分系统）发出的消旋信号 Tm，并根据 Tm 值的大小（大于 10 还是小于等于 10）选择不同的流程进行执行，当接收到空中救生系统发出的 OvSignal 信号时，主伞开启的流程被中断，整个活动结束，如图 2-36 所示。

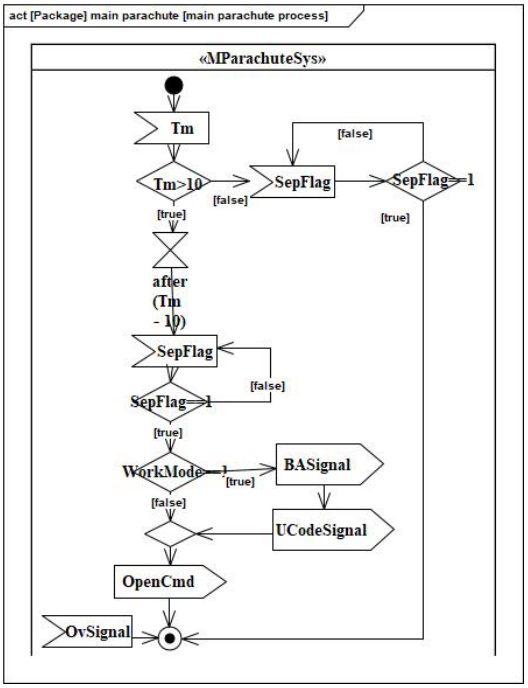


图 2-36 主降落伞系统活动图[16]

2.5.3.4 序列图

序列图是另一种可以用来说明系统动态行为信息的 SysML 图，是系统的一种动态视图。它显示了生命线的元素，描述信息的交互。当需要精确地指定实体之间的交互、系统问题域内的交互和解决方案内的交互的时候，需要建立序列图。

序列图中的主要元素是生命线。

生命线是代表交互参与者的一种元素，代表了交互中参与者的单一实例，它会与其他生命线交换数据，即消息。生命线的标识法是一个矩形，附有虚线，在序列图中显示为上下方向。虚线代表了组成部分属性的生命，生命是随着时间而改变，时间会沿着生命线向下进行，先发生的事件会显示生命线中比较高的位置，而后发生的事件会显示在比较低的位置。

生命线上可以出现 2 种类型的事件：消息发送事件、消息接收事件。生命线创建的事件消息代表了发送生命线和接收生命线之间的通信。

- 实三角线实线：——▶ 表示信息传递是同步的。
- 虚三角线实线：——→ 表示信息传递是异步的。
- 虚三角线虚线：- - -→ 表示信息回复。

图 2-37 是一摄像机控制系统序列图。有两条生命线：左侧为安全监控操作员，

右侧为安全系统。描述了从安全监控操作员选择编号为 CCC1 的摄像机开始，到从安全系统获得当前信息 Moving 的整个序列过程。

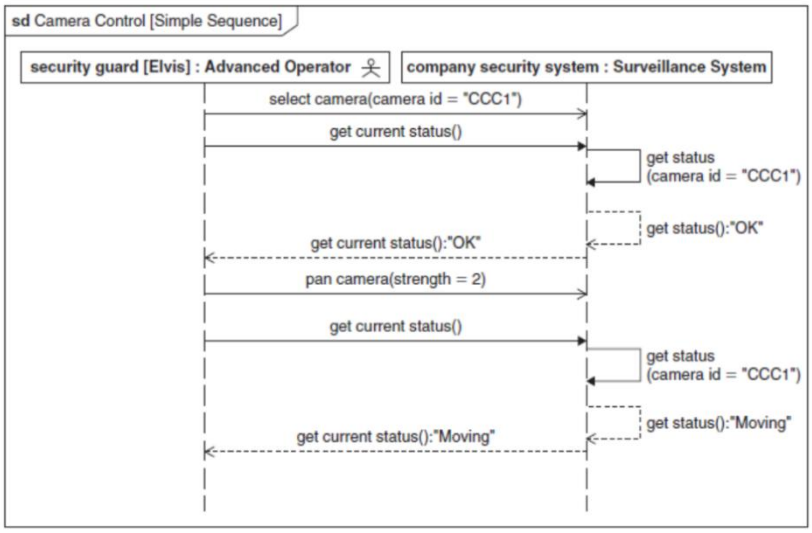


图 2-37 摄像机控制系统序列图[15]

2.5.3.5 状态机图

状态机图是能够用于说明系统动态行为信息的三种 SysML 图之一，它关注的是系统中的结构状态如何根据随时间变化而改变的过程。

状态机图显示各种各样的系统状态，可以指定四种类型的事件，以及在运行系统中状态间转换的触发条件，从而对系统的行为做精确、清晰的说明。

状态机图中的主要元素有：

- 状态：用矩形表示，每个状态有入和出的行为，状态之间由有向箭头表示，边上有触发条件。
- 初始状态：表示状态图中的开始点，符号为●
- 终止状态：表示状态图中的终止点，符号为◉

图 2-38 规范了监视系统状态迁移过程，从空闲（idle）到初始化(initializing)，若初始化成功(OK)则转移到操作状态(operating)，否则进入到检测状态(diagnosing)。若申请关闭则进入关闭状态(shuting down)。进一步地，若验证成功则输出关闭摄像机动作，然后进入空闲状态(idle)。在关闭电源后进入终止状态。

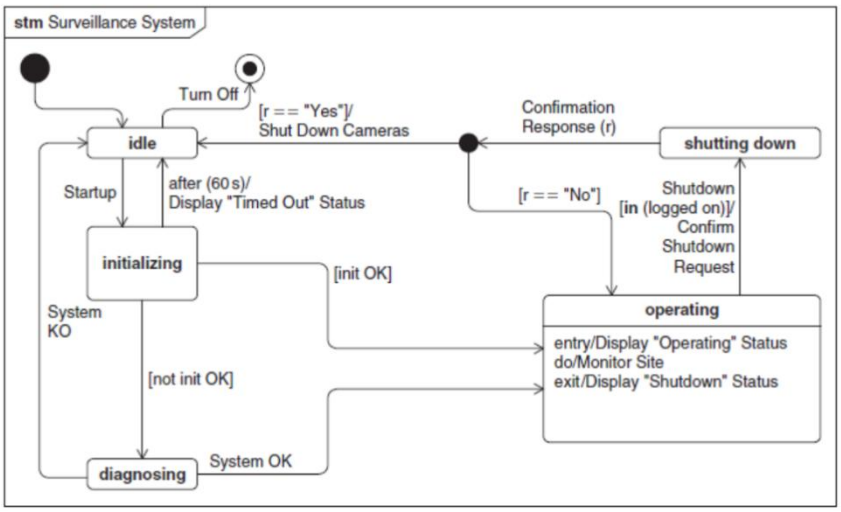


图 2-38 摄像机监视系统状态机图[15]

2.5.3.6 用例图

用例图可以显示系统类型的元素和关系，说明系统提供的服务信息，以及需要服务的利益相关者的信息。用例图是系统的一种黑盒视图，很适合作为系统的情景图。用例图应该在系统生命周期的早期创建。系统分析市可能会枚举各种用例，然后在系统概念和操作的开发阶段创建用例图。

用例图主要显示两个内容：系统提供的外部可见服务（即用例）、以及触发和参与用例的执行者：

用例是系统将会执行的一种服务、一种行为，因此用例名称总是一个动词短语。用例会捕获系统利益相关者之间关于系统行为的契约，把那些不同的场景搜集在一起。

执行者是一个人或者一个外部系统，执行者和系统之间存在接口，分为主执行者和次执行者。触发用例的执行者叫做主执行者，参与用例中的执行者叫做次执行者。

图 2-39 描述了机动车有关持有者的用例图。持有者包括驾驶员和乘车人，驾驶员是驾驶汽车者，而乘车人可以进出机动车，同时也可以操作机动车辅助系统。辅助系统包括温控系统和娱乐控制系统。

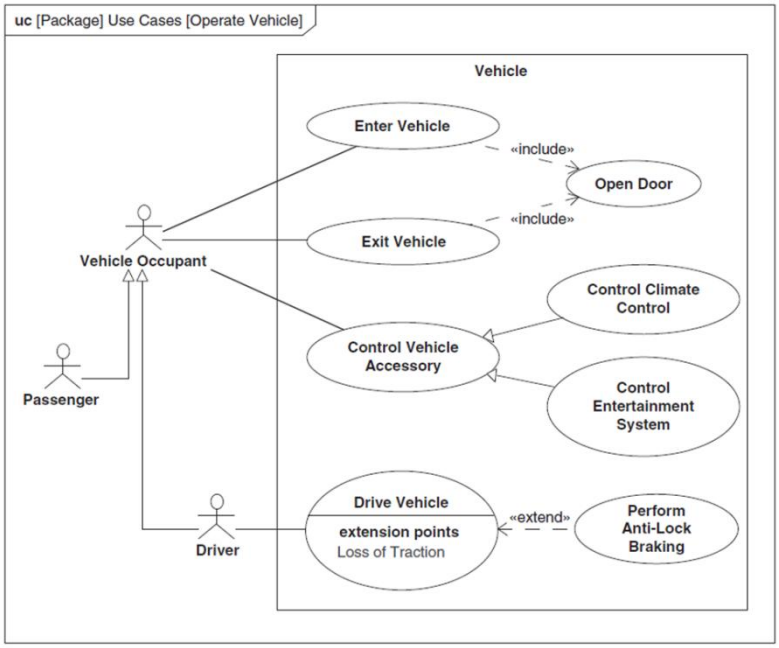


图 2-39 机动车持有者用例图[15]

2.5.3.7 参数图

参数图，包括约束图，是一种独特的 SysML 图， 用来说明系统的约束。 这种约束一般用数学模型的方式表示， 决定了系统中一些列合法的值。使用块图来建立参数图和约束图。只有参数图能够向利益相关者传递这些数学模型。

SysML 把等式和不等式建立为约束模块，以指定模块的值属性的固定关系。当需要显示不同约束表达式中约束参数之间的绑定关系时，需要建立等式或不等式的复合系统，建立参数图。

参数图的类型缩写是 par。 模型元素可以是 block 或 constraintBlock。

par [model element type] model element name [diagram name]

图 2-40 是来自文献[15]的例子，描写了能耗参数情况。模块元素名称是 Power Consumption，模块元素类型是 ConstraintBock。图中有两个 Bock，Bock pe 满足 Joule' Law（焦耳定律），Bock ps 是能量求和(Power Sum)，关联着多个构件需求(Component demands)。模块值属性包括电流 A、电压 V 和电能 W。

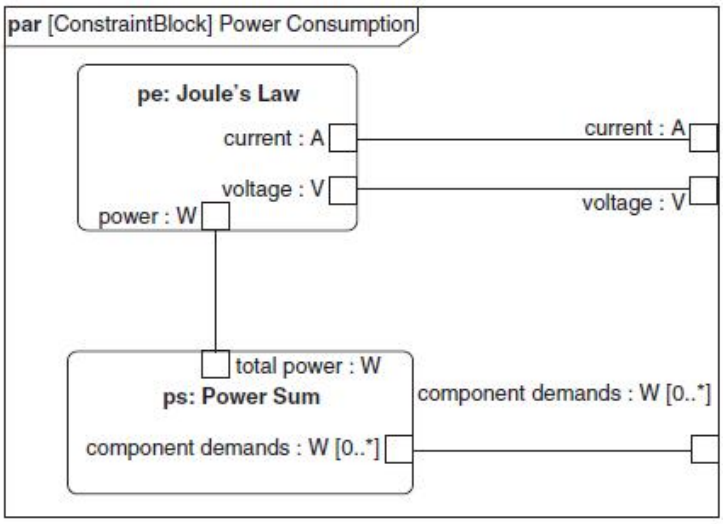


图 2-40 能耗参数图[16]

图 2-41 是一自由落体运动的参数约束图，规范了物体自由落体运动时的下落距离 d、速度 u 和加速度 a，以及时间 t。这些变量值的单位，如距离 d 的单位属性是米 m，时间 t 单位属性是秒 s，物体自由下落运动的加速度 a=9.8，是距离对时间的二阶导数。下落距离 d 是时间 t 的函数， 定义为 $d=u*t+(a*t^2)/2$。在初始时刻 T0 时，规定 t=0 时 d=0。

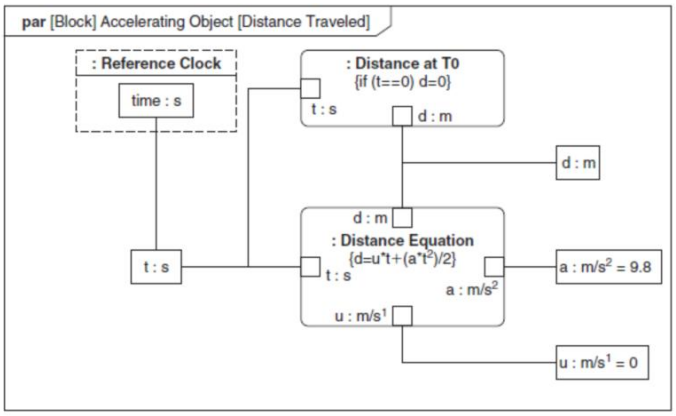


图 2-41 加速运动物体的参数约束图[15]

2.5.3.8 模块图

模块图是一种最常见的 SysML 图，它可以显示不同类型的模块元素及其关系，以说明系统结构的信息。模块定义图中的元素叫做定义元素。定义元素形成了系统模型中其他内容的基础。定义元素的重要性体现在元素之间的结构关系--关联、泛化和依赖。使用这些关系，通常会创建系统的分解和类型的分类。

当需要需求分析、需求定义、架构设计、性能分析、测试用例开发、集成都需要建立模块定义图。

图 2-42 描述了返回舱降落伞系统组成模块，包括伞舱盖弹射分离系统、主伞状态监测系统、主降落伞系统、备用降落伞系统以及空中救生系统。

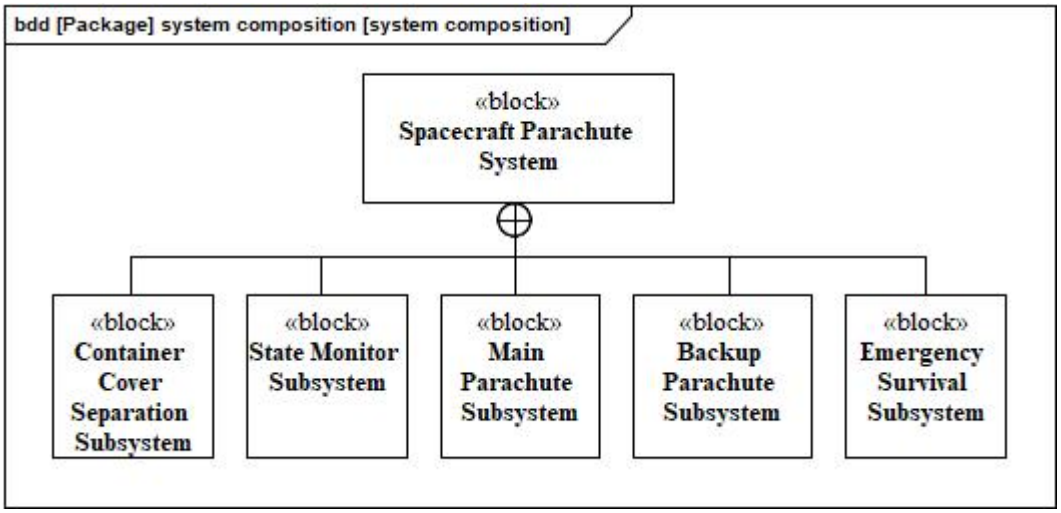


图 2-42 返回舱降落伞系统组成模块图[16]

2.5.3.9 内部模块图

内部模块图与模块定义图非常密切， 在内部模块图中显示各种元素来说明系统结构的各个方面，对模块定义图表达内容的一个补充。内部模块图唯一允许的模型元素是模块。它的外框是代表系统模型某处定义的模块，在外框中可以显示模块的组成部分属性和引用属性。

当需要显示模块的合法配置——模块属性之间特定的一系列链接，需要建立内部模块图。

图 2-43 是主降落伞的交互内部模块图，描述了主降落伞系统接收来自伞舱盖弹射分离系统发送的信息 SepFlag 以及来自主伞状态监测系统发来的过载信息 OvSignal，同时信息 BASignal 给备用降落伞系统。

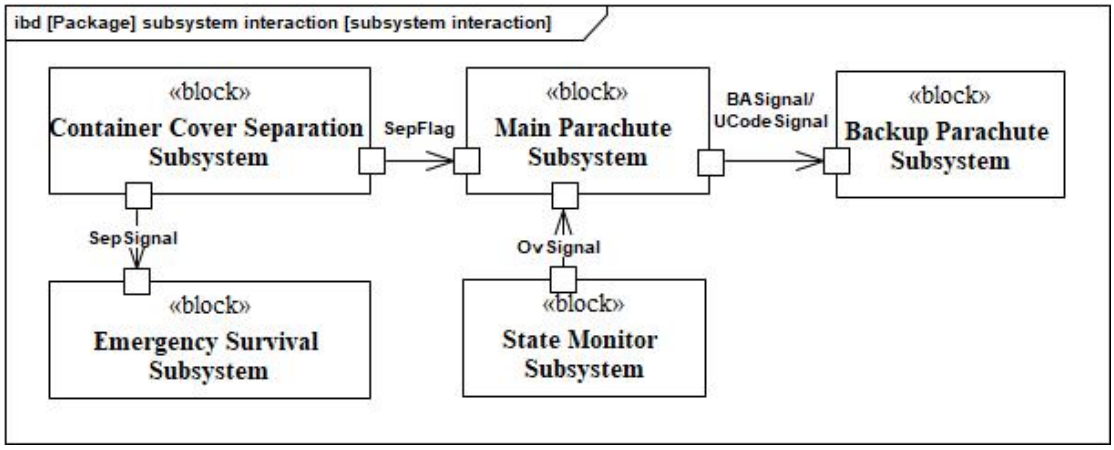


图 2-43 主降落伞交互内部模块图[16]

返回舱降落伞系统在执行过程中接收外部设备——遥感装置、过载控制器、GNC 以及雷达的信息。这些信息的传输方式可以通过外部总线或者是网络通信协议环境中各种实体的信息。图 2-44 描述了返回舱降落伞系统与 4 个的信息输入输出关系。

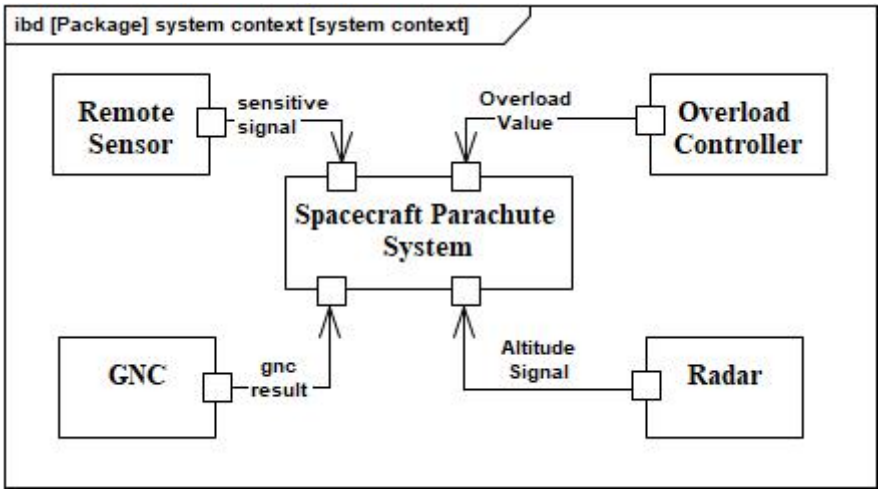


图 2-44 主降落伞信息输入输出关系内部模块图[16]

第 2.6 节 本章小结

本章介绍了智能嵌入式系统的一般建模方法和建模语言。系统建模其基本目的是使用规范的语言（建模语言）来规范系统的功能和性能，其优点是不会产生对系统需求产生歧义，同时系统开发过程的各个环节都能统一理解和掌握。另一方面，系统建模也为系统的仿真提供基础。

有限状态自动机一般用来对简单的离散控制系统进行建模，数据有限状态自

动机用来对数据驱动的离散控制系统进行建模，混成自动机可以对带有连续数据驱动的控制系统进行建模。

在本章定稿时，软件学报发表了中科院软件所王淑灵等人的可信系统性质分类和形式化研究综述文章[17]，系统地介绍了计算系统的形式化验证方法与工具，其中介绍了混成自动机，使用了共同的例子—水缸系统。

习题

2.1 建立时间控制路灯有限状态自动机模型

时间控制自动路灯是指灯的开和关是时间驱动自动控制的路灯，建立晚上 6 点钟开灯、早上 6 点钟关灯时间控制路灯有限自动机模型。

2.2 建立光控灯系统的有限状态机模型

光控灯是一种智能灯光控制系统，灯的开和关依据灯所在位置光照度变化而改变。当光照度低于 x 勒克斯时灯处于开状态，当光照度高于 y 勒克斯时灯处于关状态。这种光控灯系统可适用于路灯控制，也可以用于楼宇灯自动控制，以及汽车外设灯自动控制，查阅资料确定路灯控制光照度的阈值，并建立相应的自动机模型。

2.3 建立声时混合控灯的有限状态机模型

在居住楼道里经常会遇到声控灯，当有声音时灯的开关由关闭状态转换成开着状态，使灯保持灯亮 2 分钟，然后转换成关状态。建立此声时混控灯自动机模型。

2.4 建立下面两题的 Moore 型和 Mealy 型自动机

(1) 单部 5 层电梯控制系统。

(2) 饮料售货机可售 1 种饮料：可乐。每瓶可乐 4 元，可接收现金 1 元与 10 元，现金找零。

2.5 设计下面三题的 Mealy 型 FSMD 模型

(1) 单部 10 层电梯控制系统。

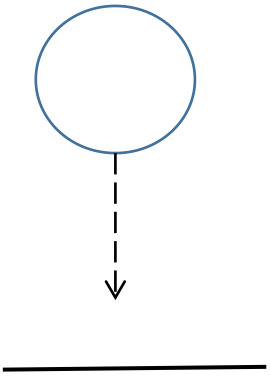
(2) 南北东西两个方向交通路口红路灯正交控制系统：南北方向直行绿灯 40 秒，东西方向直行绿灯 30 秒，在直行时可以左转，右转始终是自由的。正交控制系统是指南北方向为绿灯时东西方向为红灯，南北方向为红灯时东西方向为绿灯。

(3) 饮料售货机可以售 2 种饮料：茶和水。每瓶茶售 3 元、每瓶水售 2 元；

线上支付。每次可以购买 1-3 瓶饮料。

2.6 汽车自动停车系统的混成自动机模型

汽车自动停车系统按照汽车分成三个阶段进行，第一阶段是匀减速行驶，减速加速度是 dv/dt=-1.35（米／秒平方），当车速到达每小时 20 公里速度时，进入第二阶段，第二阶段也是减速行驶，减速加速度是 dv/dt=0.09t-4.36（米／秒平方），当车速到达为零时，汽车进入第三阶段，停车。建立汽车自动停车系统的混成自动机模型，并画出混成自动机演化过程，车速初始速度为 100 公里/小时。



2.7 弹跳球运动模型

让球体在高度 h 处放下做自由落体运动，当落地时受到下落力作用，球会弹起，速度损失 20%，到最高处又会受到地球引力作用做自由落体运动，这样反复落-弹运动，直到球落地不再弹起为止。建立弹跳球运动系统的混成自动机模型，并画出 h=100 厘米时弹跳球运动演化过程。

2.8 建立汽车自主防撞系统的混成自动机模型

汽车自主防撞系统是在汽车行驶过程中自动感知前面是否有静止障碍物、行人和行驶中的汽车等，这些统称为障碍物。若存在则汽车自动采取减速、停车或避开措施，避免发生碰撞事故。自主防撞系统引起汽车行驶状态的转换，汽车行驶状态包括：

行驶状态：汽车感知前面没有障碍物或者与障碍物间距大于自主防撞系统计算出的安全距离 A（70 米）时，系统控制汽车进行（正常）行驶状态，行驶速度为每小时 80 公里；

减速状态：自主防撞系统检测到障碍物并且障碍物与车之间的距离小于安全距离 A 大于安全距离 B（50 米）时，汽车进入减速状态，速度=-3.37t^2+22.22（米／秒），减速加速度=-6.75t（米／秒平方）；

制动状态：若障碍物出现并且汽车与障碍物的间距小于安全距离 B（50 米）大于安全距离 C（20 米），自主防撞系统则进入紧急制动状态，减速速度=-4.05t+23.44（米／秒），减速加速度=-4.05（米／秒平方）；

停车状态：若汽车与障碍物的间距小于安全距离 C（20 米），则自主防撞系统进入停车状态，速度=-6.5t+19.88（米／秒），减速加速度=-6.5（米／秒平方）。

汽车在行驶中显示与前面障碍物距离，若前面没有障碍物，则显示距离为符号∝。依据这些参数建立汽车自主防撞系统的混成自动机模型，行驶状态、减速状态、制动状态可以相互转移到达，但行驶状态不能直接到达停车状态，停车状态为终止状态，行车状态为初始状态。

2.9 调查业界大型智能嵌入式系统 SysML 建模典型案例。