

图文显示的编程规则

TM630



贝加莱工业自动化
Perfection in Automation
www.br-automation.com



前提

培训模块: TM600 – ASiV 的基础

软件:

-

硬件:

-

目录

1 · 简介	3
1.1 目的	4
2 · 路线图	5
2.1 项目阶段	5
3 · 说明书	7
4 · 软件设计	10
4.1 默认的项目	10
4.2 可视化的结构	11
4.3 操作概念	19
4.4 变量和数据点	21
4.5 可视化运行行为	24
4.6 可视化程序设计接口	25
4.7 用于维修和起动的屏幕页	27
4.8 数据和数据管理	28
5 · 集成可视化	32
6 · 项目的维护和组织	33
6.1 项目的变化	33
6.2 软件分配和储存	34
6.3 把项目让渡给客户	35
6.4 文档	36
7 · 小结	38

1、简介

在不同的情况下，我们会面临同样的问题——“应该从哪里着手开始新项目”？

尤其是在没有项目指导或项目指导没有具体指明时，这个问题更为重要。这个培训模块提供了一些建议，帮助用户——大多针对可视化项目——克服最初的障碍，并最终节省时间和费用。

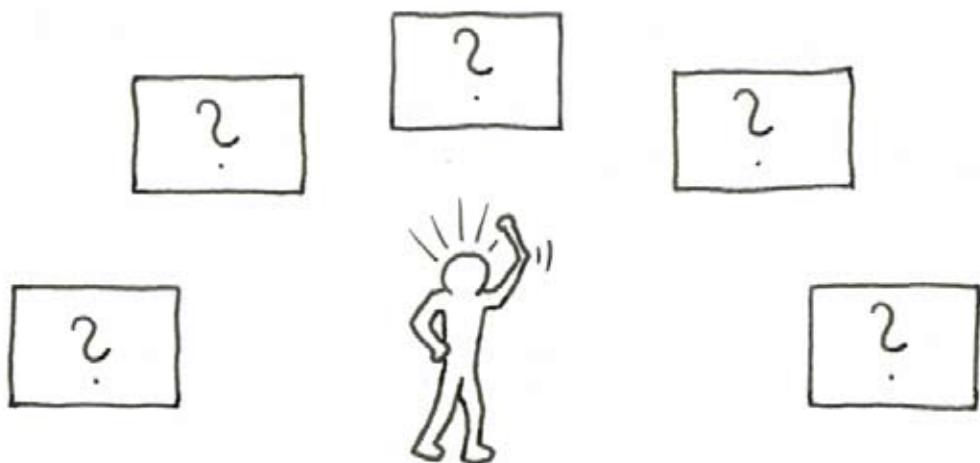


图. 1可视化程序设计指导

也许只有专家才能找到一些对设计新的可视化项目有用的信息。

在创建可视化应用时，你必须投入到可视化目标组的角色中（操作员，维修技术人员，高级程序员）并理解面向任务的过程。只有这样你才可能在遇到任何要求时都能找到最佳的解决方案。

简介

1.1 目的

使用这个培训模块中的信息来设计自己的可视化项目。

这可以防止“直接进入”编程模式，而此时程序员并没有充分的计划或足够的知识。



图. 2 综述

2、路线图

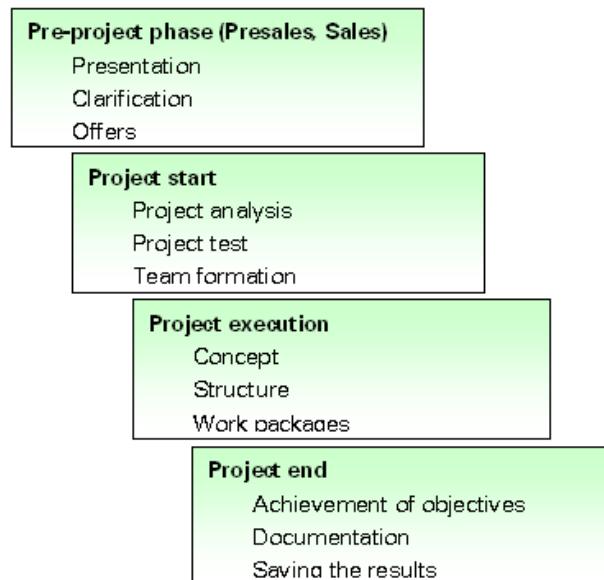
路线图保证了项目有组织的执行并且不需要监视。

“当然可视化应用程序必须在昨天就结束，假定机器在前天是可操作的。”

2.1 项目阶段

在开始执行项目前，首先要清楚客户的要求。

下面的图像提供了路线图的总貌。

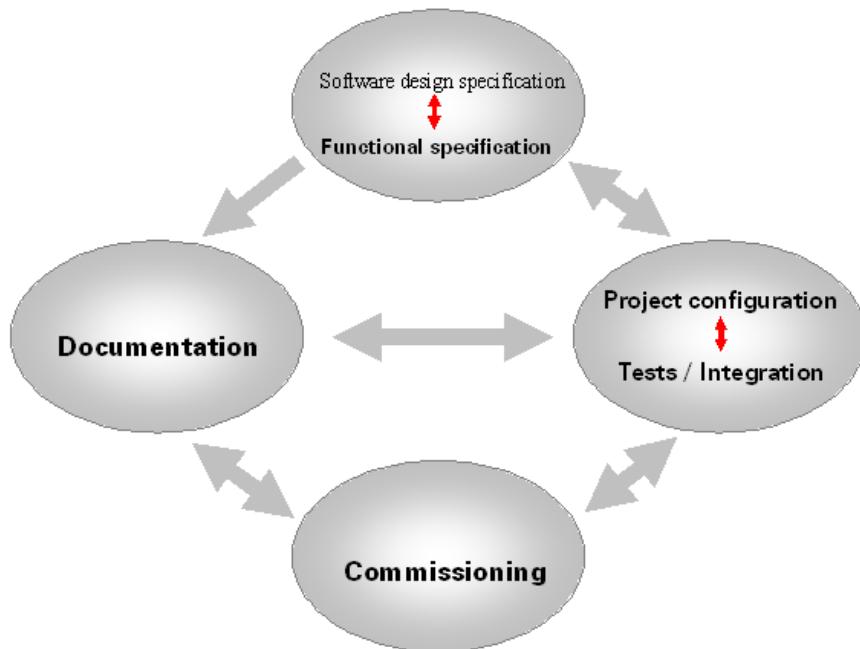


这个培训模块处理的是项目执行阶段。

路线图

项目执行阶段可以被粗略的分为四部分，各个部分之间有着相当大的相互作用：

- 创建软件设计说明书 估计所需的全部工作 功能说明书
- 图象、数据管理等的设置
- 不同项目状态的测试和集成
- 机器的执行
- 文档



3、说明书

不管使用什么可视化设备，客户的要求和说明书或软件设计说明书将决定其基本内容。

花费时间 [%] 实际		花费时间 [%] 目标	
说明书	0-20	说明书	30
项目配置	80-100	项目配置	20
测试	20-0	测试	50

如表中所示，项目中有1/3的时间用于创建说明书或软件设计说明书。这会缩短项目配置时间，因为之后项目很少做改动。

但事实总是有与计划不同，因为这个时间对于“普通的”项目配置一般是不够的。程序员应该考虑到在项目配置阶段详尽的说明书是否真能节省时间。

备注:

在创建说明书时，要避免定义每一处，应在必要信息和程序员的自由之间应找到妥协点。

说明书

在建立可视化说明书时，要回答以下的问题：

- 使用哪类可视化设备（价格范围，运行成本）？
- 使用哪个可视化设——嵌入式、远程显示设备，等等？
- 是否与更高等级的系统相连接？
- 需要哪些标准和安全功能？
- 数据转换、数据格式和赋值——是否取决于使用的硬件？
- 谁将操作可视化应用？
- 需要操作什么？
- 哪些信息是重要的？
- 使用哪种操作概念（按键、触摸等）？
- 如何构成页，显示哪些元素？
- 主页面是什么，需要哪些和多少子级别？
- 使用哪种颜色和形状？
- 在页面中有导航功能吗？
- 在页面这间有导航功能吗？
- 谁可以操作哪些页面（密码等级）？
- 如何对页面分组（页面改变、菜单）？
- 需要那些字体和字号？
- 需要什么语言，需要多少种语言？
- 使用哪些文本和图形？
- 密码管理？

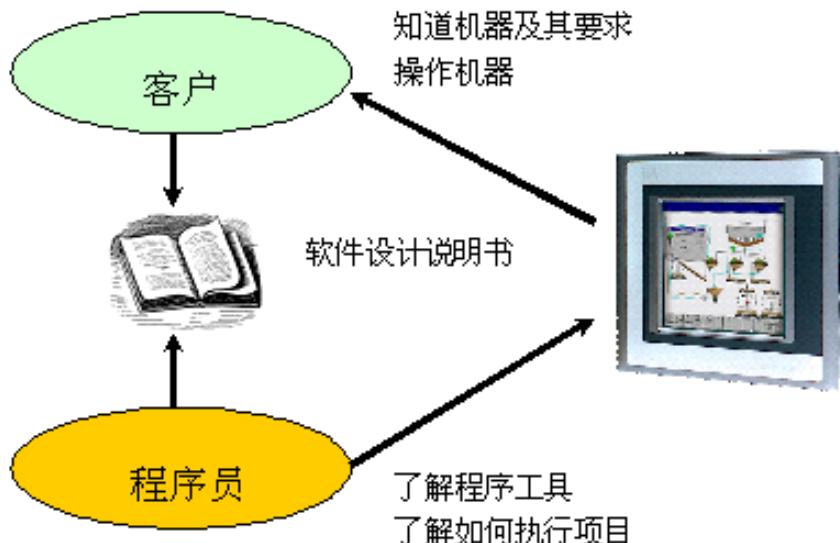
这些信息是创建可视化的基础。

备注：

在客户开始安装和设置页面之前，必须创建几个例子给客户演示。客户选中的例子可用作工程的模板。

客户也必须清楚，以后在工程中所做的任何改动都需要重新编辑为改动所创建的每一页。

说明书必须在客户的协作下才能确定，因为用户或程序员比较熟悉可视化软件和创建可视化的编程环境。



这是使用现行的手段满足并实现客户要求的唯一途径。

备注:

当整理软件设计说明书时，程序员应该已经在考虑项目的可视化问题了。

如果程序员这时还在学习如何使用工具来创建项目的话，那么他还不能评估正在使用的可视化软件是否满足各种要求。

4. 软件设计

对于正在处于编程阶段而非理论阶段的项目来说，这种说法"it just sort of happened"是很常见的。

在最初的项目建立之后，将很多改动"汇集到一起"要比在开始构建项目时，在执行和维护方面难的多。

4.1 默认的项目

新项目一般以空白页开始，然后将所需的组件如位图和文本添加到项目中。由于每个项目的结构不同，所以每次会在不同的位置添加那些元素。

如果在相同的结构基础上开始创建项目，可以使用默认的项目或模板。

项目相似或完全不同，这并不重要——当创建一个新项目时，好的基础会帮助你节省时间。

默认项目中的元素

- 数字和字母输入的键盘和触摸的设置
- 单位组（比例和格式）
- 位图和位图组
- 字体和字号
- 默认页面、页眉、页脚、模板
- 风格

备注:

一旦创建好“新项目”的基础，那么每次开始项目时都会使用它。客户必须认可默认项目中的所有元素。

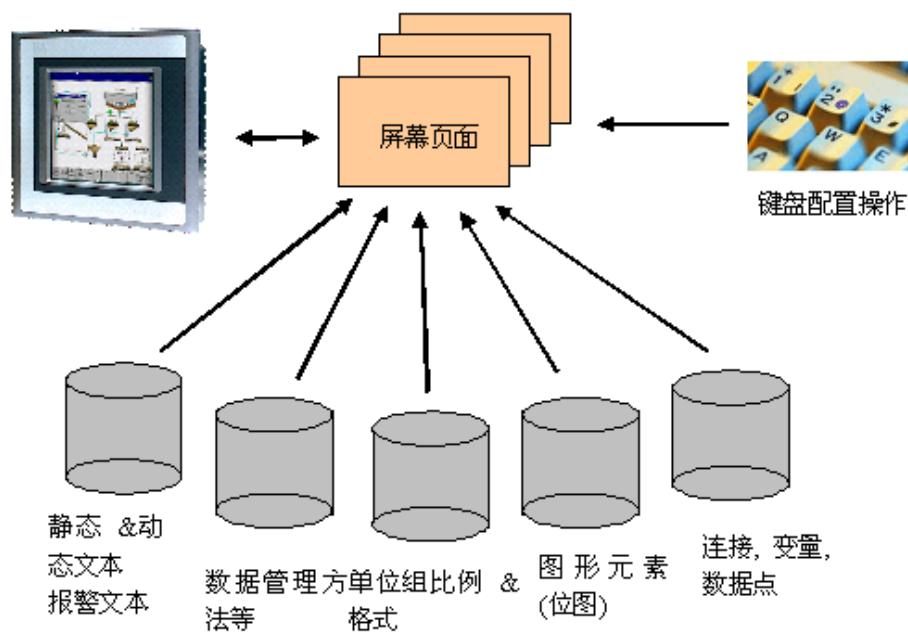
所有的屏幕页必须要重新修改，例如页眉的大小会因像素或字号以及字体的改变而改变。



4.2 可视化结构

可视化项目分成不同的项目组件。这些组件的分配和设置对每个可视化一般是不同的。

可视化有可能包含的组件:



这些组件可以根据需要扩展。不包含在可视化中的功能可以通过附加的程序来实现。

4.2.1 屏幕页面设计

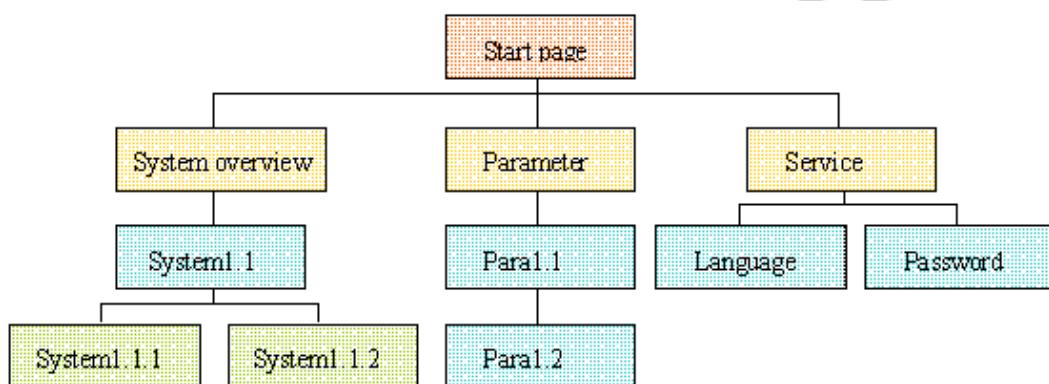
屏幕页面是可视化的的主要组成部分。在设计屏幕页面时，必须考虑一些规则：

4.2.2 页面分支和分组

当你知道了每个屏幕页面的内容后，就可以将页面分组，并基于以下的结构：

- 过程流程
- 用户等级
- 服务和维护等级

树型结构用于确定理论上的菜单设计和页面变换。

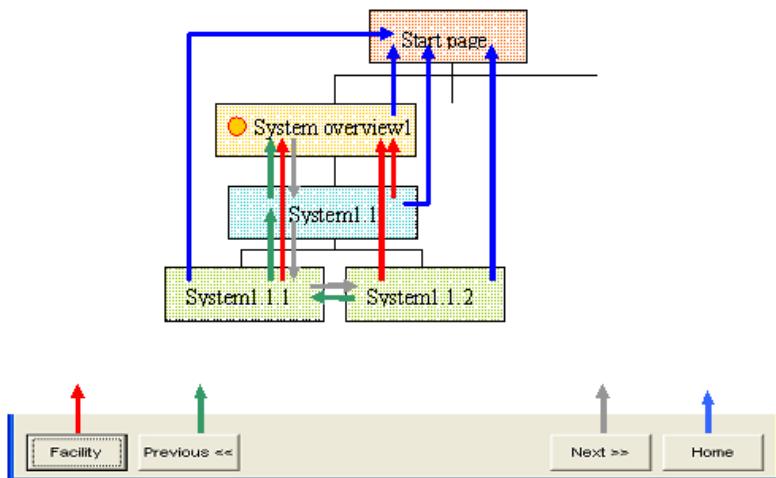


在这个例子中，项目页面列表中的页面分组看起来是这样的：

0000_StartPage
0100_SystemOverview
0110_System1.1
0111_System1.1.10
112_System1.1.20
120_System1.20
200_Parameter0
210_Parameter1.10
220_Parameter1.20
300_Service0
310_Language0
320_Password0
500_AlarmOverview0
510_AlarmHistory

在创建页面分支时应遵守以下规则:

- 应该总是从屏幕页面的相同位置（屏幕页面上的触摸按钮或页面转换按键）调用页面。
- 当页面转换到下一级页面时，用户应能返回主级页面而无须通过一些页面导航。



在这个例子中，页面等级之间的导航会使用不同颜色箭头来显示（从等级“1”开始）。

- 红色箭头：从任何等级低的页面都可以到达等级“1”
- 绿色箭头：从最低等级，这个箭头可以返回同一等级内。从较低等级的第一页，这个箭头可以返回到上一个等级。
- 灰色箭头：在相同的页面等级中，这个箭头可以走到下一页。在等级的最后，这个箭头会走到下一等级的第一页。
- 蓝色箭头：这个箭头可以从任一等级返回起始页。

当然对于页面变换有很多不同的概念。这个例子只描述了页面更改次序的一种可能。

软件设计

页面分组建议:

- 在页面名称前分配分组号。
- 在分组号后留有空间，以便以后向组中添加页面。
- 在每页上显示页面等级，以便得知当前正在显示的是哪一等级。

改变页面:

一般情况下，用键盘或触摸设备来改变页面。目标页面总是直接设置到按键或触摸动作上。这里不推荐由控制变量来改变页面，因为每次改变页码时应用程序都必须改变。

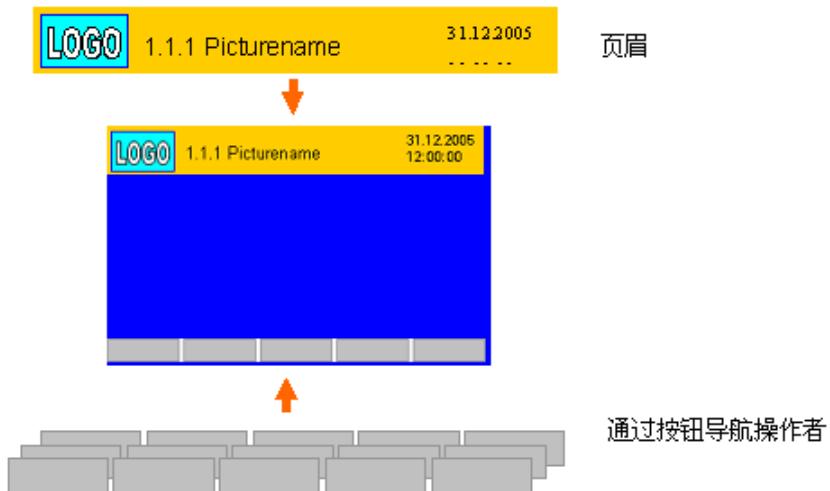
警告:

当控制程序初始化某个动作时(例：发生报警时)，或使用有密码的导航页面时(在检查密码后才改变页面)，使用变量改变页面也是必要的。

可视化中也提供页面改变时的延时功能（例：如果用户在某页上执行了一个动作，而此动作不能中断）。

4.2.3 分层

分层法可以在一层上放置相应的页面元素，并按需要经常使用这些元素。在不同的可视化包的配置、功能性和用法上存在区别。



分层法的优点:

- 用于多个页面中的图象信息只需创建一次。
- 以后所做的改变只需在一个位置执行
- 根据设置，在运行中层可以设置为可见的或不可见的。
- 分层法用于执行不同的对话框
- 控制机器选项的可见性。

备注:

只要支持层的使用，就应该明确地使用各层。因此，在创建屏幕页面之前必须决定哪些层显示哪些屏幕信息。

4.2.4 可视化文本

可视化文本包括静态文本和动态文本。

静态文本是屏幕页面的一个固定部分，如显示元素的描述文本。

动态文本根据连接文本元素的变量来改变实际的文本。

备注:

如果有可能，静态文本和动态文本应该在易于管理的相关功能的文本组中结合使用。

优点:

- 用于多个屏幕页面上的文本只需创建一次。
- 以后的文本改变只需在一个位置执行
- 不同语言的文本只需分别翻译一次（节省成本）。

相关语言文本

如果有可能，同一种语言的文本应该在各自的文本文件中管理。

优点:

- 在建立工程时，只有使用的语言文件和要翻译的语言必须翻译。这就使用于多个位置的文本只需翻译一次。
- 只有已指定过的语言才能传送给可视化。其它语言可作为选项添加。

4.2.5 图形元素

管理图形元素

为了改进可视化的总貌和组织，相似的图形元素，如位图，应该分到逻辑组或单独的子目录中。

活动的图形对象

在可视化中，活动的对象或不断变化的元素个数应该保持在最小。这些对象会分散操作者对更重要信息的注意力。

备注:

如果在屏幕中有多个循环变化的对象（例：闪烁的报警符号或光标），那么确保所有的元素以相同的闪烁频率变化。

4.2.6 使用标准

对于程序员编程特别是在Windows中编程时，可以延伸标准输出对话框的大小（信息栏，文件打开对话框等）。但需要考虑几件事情：

- 标准对话框中的文本总是显示安装库中的语言。一般来说，在系统运行中这种语言是不可切换的。
- 这些对话框中的按钮大小不一定合适。
- 并不是总能输出UNICODE字符。

为了避免这些问题，可视化中所有对话框都放在独立的屏幕页面或屏幕等级上。

4.2.7 每个元素都有名字

当在项目或组件等级的可视化中添加元素时，系统赋予每个元素一个默认的名字。这个名字显示了元素的特征，但如果几个相同的元素，那么靠名字区分它们是困难的。

这会使第二个人在编辑相同的项目时变得更加困难。

使用默认的名字工作	使用带有描述性的名字工作
Visualization	Vis_MainTerminal
Image1	000_StartPage
InputField_1	txtSetPassword
OutputField_1	txtActPasswordLevel
Image2	010_Parameter
InputField_1	txtSetCoolingDelay
InputField_2	txtSetCoolingTemp
OutputField_1	txtCoolWaterTemp
Button_1	cmdPage_AlarmHistory

正如这个例子所示，元素名可以改变成描述其实际用法的名字。

备注：

必须给可视化对象赋予有意义的名字——对控制器编程时，使用客户指定的编码规则。

使用默认的名字工作 使用带有描述性的名字工作

使用默认的名字工作	使用带有描述性的名字工作
Private Sub Command1_Click() End Sub Private Sub Command2_Click() End Sub	Private Sub cmdStartHeating_Click() End Sub Private Sub cmdStopHeating_Click() End Sub

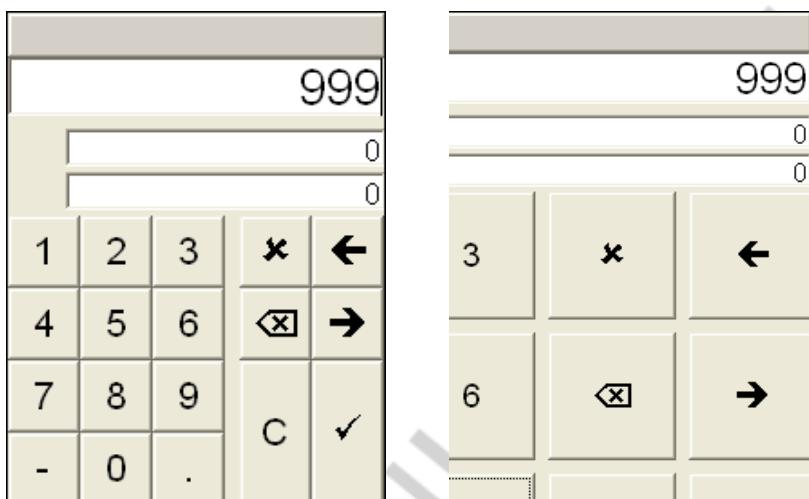
4.3 操作概念

可视化硬件决定了如何操作系统。

使用硬件时，有些操作方面的限制：

触摸：在系统中，不能同时操作几个按钮。触摸控制器返回的总是平均位置，除了使用矩阵触摸系统外，此时整个触摸表面化分成一个固定的矩阵。

操作元素的大小：操作元素的设计和大小应与用户的要求匹配。客户应该清楚如何操作系统。如果系统主要由带手套的人来操作，那么程序员不应假设操作者会脱掉手套来操作可视化。



在这些情况下，操作元素（按钮、数字和字母触摸输入区域、触摸板等）的大小应按需要来设计。

这些事情需要从一开始就与客户一起规划。在工程后期，每一处改变都需要花费大量的工作。（每一页都需要改变）。

软件设计

键盘：应该检查一下键盘是否支持同时敲击多个按键。

按键/触摸的实时性能：在过程工程中，当按下按键或触摸区域时会开始一个动作，当释放按键或触摸区域时会停止这个动作 = 点动按键模式。这需要高速的响应时间(< 50ms)。

并不是所有的可视化应用都能保证这样的响应时间。因此，推荐使用硬接线的键盘（硬件设备）。

鼠标：对于机器的操作（现场可视化），鼠标的使用并不是非常普遍。然而，在过程控制系统和控制站或主办公室的可视化中，鼠标的使用是极其普遍的。

备注：

为了在可视化中执行不同的操作原理（例如鼠标和触摸），必须考虑可视化有些限制——例如不能实现快捷菜单（鼠标右键点击）。

4.4 变量和数据点

在建立可视化时，最重要的问题之一是配置变量或数据点。

变量 (PV)

定义控制器的地址

数据点 (TAG)

如果可视化允许输出附加属性（读、写、事件、比例、限制、格式、转换、读取周期等），那么变量作为数据点来考虑。

变量及其属性/特征的恰当设置可以在很大程度上增强可视化的性能。

属性	动作
事件 / 获得	<p>在默认情况下，可视化中的变量是由控制器循环读取/获得的。读取的实时变量越多，循环更新的显示就越慢。如果要操作某个事件，那么就必须中止循环通讯中的有些负载，因为控制器处理检查任务时，事件变量值就会改变。因此，剩余的变量可以更新的更快更频繁。</p> <p>例子：</p>  <p>在这个例子中，变量 "PV1" (必须以快速的数据变化来显示) 和另一个变量 "PV2" (以慢的温度变化来显示) 为每秒读取一次 (读取控制器中的值)。</p> <p>如果变量"PV2"用作事件变量 (其它变量)，那么获得的变量可以按要求来读取。</p>
循环读取(刷新)	<p>变量的刷新时间决定了控制器读取变量的时间有多快 [ms]，或控制器执行事件监测的时间有多快。改变变量的刷新时间，从快速到慢速或仅刷新一次，会减小循环通讯的负载 (例：温度变化——刷新值=2000ms)。</p> <p>例：</p>  <p>在这个例子中，所有变量以相同的读取周期来读取。 (上图)</p> <p>如果将“慢速”变量的读取周期改为快速，那么它也能以“快速”变量的形式频繁地读取。</p>

软件设计

在可视化中，哪些变量需要“主动地”读取：

- 显示在页面上的所有变量或与页面元素相连接的所有变量。
- 必须在后台的报警系统或趋势系统读取的变量。这种数据通常随后就会显示。
- 不必改变变量值就可以将变量切换为“不激活”状态的变量。当页面变化时，状态改变会从“激活的 不激活的”。

带有页面元素的变量（例数字输出区域）与每个屏幕页面相连，并由控制器“主动地”读取。

页面上没有显示的变量不需要读取——这减少了通讯负载，并使变量可以在活动的屏幕页面上更快的刷新。

访问变量的方法

除了对变量的读和写访问外，“同步”和“异步”访问变量也是一种常见的区别。不同的可视化系统可能使用这些方法的条款不同，或者只支持其中的一个。

- 当使用同步访问时（读/写），程序会等待任务调用的确认。在这期间，不执行其它任何操作。在一个循环中调用多个同步功能是明确禁止的。
- 当使用异步访问时，确认的步骤稍后执行。在答复或确认中，会通知应用程序刚才的访问是否成功或是发生错误。

数据的一致性/数据访问步骤的同步

数据的一致性：当变量与输入或输出区域连接并按比例显示时，数据的一致性可以得到保证(例 INT, DINT, REAL,等)。

如果在运行中需要传送大量的数据（结构，数组），那么数据的一致性就不再有保证了。因为变量在可视化与控制单元间传送时，与控制器程序的顺序不同步。在这种情况下，应用程序必须通过同步数据交换来处理数据的一致性。

为了保持数据的一致性，必须在控制器上创建一个独立的同时写或读的图像。数据交换用触发器变量来控制（传送开始，传送完成）。

警告:

只要访问变量和变量变化是来自可视化和控制器的，那么应该同步估计不同的数值变化(0,1,2,3,...255,0)。

避免使用BOOL变量(0,1,0)来执行同步，因为变量可能由于读取周期而忽略了数据的0-1变化。

读和写变量的优先级

如果控制器和可视化硬件之间的独立通信需要在可视化上编程，那么必须确保写变量的要求(可视化 => 控制器)比读变量的优先级要高。

读变量



写变量 PV2



当有写请求时，当前的读请求必须等待，执行所有的写请求。

4.5 可视化运行行为

在运行中，必须考虑决定可视化特征的几件事情：

- 当按下页面转换键后，页面转换必须在合理的时间内执行。应记住屏幕页面上的页面元素的数量也会影响执行页面转换的时间。
- 应避免妨碍操作的行为。如果没有其它的方法（例装载数据等），那么妨碍操作时必须以适合的方式显示出来（前进的棒状图）。
- 当获得真正的数据时，屏幕页面元素中的“默认文本”才能显示在屏幕页面上。至少会显示一个“有意义的”初始值。
- 如果在启动时要求有可视化，那么可视化需要在一个单独的启动屏上初始化。在初始化完成后，用于操作系统的第一个屏幕页就会显示出来。

4.6 可视化程序设计接口

可视化包括绘制页面以及与页面元素连接的变量/数据点。

可视化不支持的功能必须由可视化提供的编程语言来实现（script，控制任务，等）。

4.6.1 出错检测

也必须评估每个功能中的返回值。这是在运行中发现并处理出错信息的唯一途径。

每个不为0的返回值必须显示在可视化的独立屏幕页面上（报警系统，单独出错记录，等）。



提示:

打开帮助后，使用<ALT> + <TAB>组合键在AS和帮助之间切换。

4.6.2 从属语言编程

当编辑文本时，确保文本只是通过它的属性（文本号）来管理而不是通过实际文本（字符串）来管理。因此，在语言转换或重新编辑文本时，才能总是显示当前语言的正确文本。

当使用UNICODE语言时（16位字符设置），在控制器和编程环境中必须使用相应数据类型（2个字节的数据类型）。

4.6.3 可视化任务

在可视化中，已显示的过程图表与控制器上的过程顺序间有着不断的相互作用。

可以通过不同的途径管理可视化任务：

- 只有一个可视化任务：可视化过程中需要的所有过程和功能都在一个任务中来管理。这种方法的缺点是任务很快会变得庞大而混乱。
- 从属页面的可视化任务：创建多个与可视化相互作用的从属过程任务。任务根据过程顺序的优先级在需要的任务等级中创建。时间不紧迫的过程在控制器的闲置时间执行（控制页面结构图等）。

应该考虑到（用户任务中的）相互作用也依赖于正在显示的屏幕页面（执行）。

4.7 用于维护和权限的屏幕页面

可视化不仅包括系统操作的“一般”画面，同样还包括一些权限画面和维护机器的画面。

一般的操作者不允许访问带有密码保护的维修和维护页面。

我们经常需要一些用于启动或将来维护任务的特殊工具。这些工具必须建立在可视化中（=工程建立时间），或已经作为独立的软件包来使用。

- 是否已经有处理这些服务的工具或现存的软件包呢？
- 这些存在的软件包可以集成到可视化中吗？如果可以—怎么集成？
- 这些产品要求软件许可吗？
- 在使用新硬件时，是否具有适应问题的潜能？
- 当在我的软件中集成这些工具时，在哪可以获得支持？



4.8 数据和数据管理

每个可视化都在运行中创建数据，这些数据为用户提供了有关过程的信息（记录、分析、存档）。

对于管理数据没有错误或正确之分。可以由用户来指定管理数据的方法，可以由程序员选择喜欢的方法。



4.8.1 管理数据



如果有可能，应该将一类数据在相同的目录中管理，使将来分析访问数据更加容易。

根据数据的类型来指定数据的名称和扩展名(例如 Trend_01012005_001.trn)

现有的报警、趋势和协议系统的名称通常是由系统决定的。

4.8.2 数据格式

根据硬件和软件不同，可以使用不同的数据形式。每种形式都有其优缺点。

在可视化中也可以使用不同的数据形式（例如初始化参数使用ASCII文件，报警和趋势记录使用二进制文件，协议系统使用XML）。



ASCII文件: ASCII文件是指那些只包含符合ASCII标准的简单文本的文件。这些文件几乎可以在所有的计算机中读出，因此容易在不同的系统之间转换。

然而它的缺点是这些ASCII文件的结构不遵循任何规则或格式。此外，在大的ASCII文件中进行搜索时要耗费更多的时间，因为必须搜索整个文件。

CSV文件是一个例外。CSV文件是一种在表格中构建数据的文本文件。缩写CSV代表"Character Separated Values"字符分隔值或"Comma Separated Values"逗号分隔值，因为每个值是由一个特殊的分隔符分离的——一般是逗号。然而分号、冒号、标签和其它字符也是常用的。这种文件形式没有官方的标准。

CSV文件经常使用扩展名.txt代替.csv，并且可以在任何文本编辑器中创建和编辑。

注册(Windows): 注册不包含文件，但是用户定义了可选项，这些选项允许用户动态地调整程序和操作系统。该信息可以包括程序的规划，不如预先定义的窗体位置。

Name	Type	Data
ab](Default)	REG_SZ	(value not set)
ab]frmMainHeight	REG_SZ	11190
ab]frmMainLeft	REG_SZ	-60
ab]frmMainPosition	REG_SZ	0
ab]frmMainState	REG_SZ	2
ab]frmMainTop	REG_SZ	-60
ab]frmMainWidth	REG_SZ	15480

注册条目是用按键创建的，这些按键是从位于更远层次上的主按键分支出来的。用户应该在练习时应特别小心，只有在绝对确定时才改动注册，因为这里的错误会影响系统的功能。

在注册中保存动态参数的缺点是客户不容易做任何改变。

在使用Windows嵌入式操作系统，并且系统分区带有写保护时，改变注册是比较复杂的。

XML文件:Extensible Markup Language, 缩写为XML，是一个创建文档的标准，格式为人和机器都能读的树状结构。XML为这些文档的结构定义了规则。各个文档的细节必须根据具体的应用实例指定（“XML应用”）。XML特别会影响在文档树内构建元素及其组织的定义方法，不同的是：XML在定义文档类型时会提供使用的规则。

数据库: 当处理大量数据时可以使用数据库。根据特殊的特性和规则来记录、整理和存储数据。

用户使用系统提供的标准函数来访问这些数据库。

当使用数据库时，它们并不是自动存储的。这就意味着对于数据库的每个新条目会需要更多的内存，即使数据被删除了。因此，必须经常压缩数据库（即重新组织数据库），并且在这个过程中不能访问数据库。

二进制文件: 与普通的文本文件不同，二进制文件包含非字母的字符，并可以使用任何字节值。二进制文件经常用于保存数据而不是保存文本。

4.8.3 数据保存和存档

在存档数据时必须考虑到关于设置的一些问题：

- 要处理多少数据？有足够大的存储单元吗（CF卡等）？
- 以后将会如何处理数据（外部工具）？
- 需要特别的软件许可吗？
- 这些数据要存档多长时间？
- 数据会储存在外部吗（网络、储存介质）？
- 哪些事件必须记录下来（报警、趋势、协议等）？
- 数据将如何记录（循环的，事件-控制的，当改变时）？
- 会基于语言评估数据吗？如果是这样的话，那么必须存储一个参考（文本索引）而不是实际文本。

一旦确定了这些条件，那么数据的从属关系和数据形式就可以确定了。

4.8.4 兼容性

如果在项目设置中或启动后必须改变数据记录的格式，那么必须确保“老的”数据仍然兼容（即这些数据仍然可以在可视化中使用——向上兼容）

如果老的数据不兼容，那么必须改变这些数据以便以后分析。

可以使用可视化软件或外部工具实现数据转换。

备注:

如果在工程设置时，程序员已经知道将要改变数据格式（例如由于将来的扩展），那么文件中的版本指示对将来指定不同的格式是非常有用的。

集成可视化

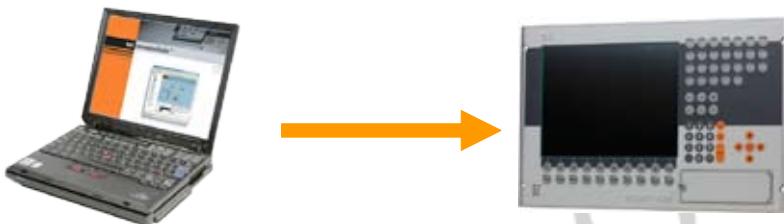
5. 集成可视化

产品的集成（即测试）是项目设置中一个重要部分。如果有可能，测试应该咨询客户的意见。

在制定可视化的软件设计说明书的一开始就应该指定测试步骤。

备注:

可视化应该集成到一个重新安装的目标系统和开发系统上。这是保证测试环境与客户使用的系统相协调的唯一方法。



开发

运行

可视化的初始测试是在办公室的环境中执行的。在成功完成这些测试后，其它的测试将在系统上执行。

必须执行的测试有哪些:

- 测试与控制器项目的通信
- 所有的变量都正确显示和运算了吗？
- 记录和输入限制都正确吗？
- 正确设置了所有的按键功能吗？

为了完成初始测试，可以在控制器上创建一个包含所有需要变量的虚拟项目。

为了测试可视化的项目次序，必须使用仿真程序测试可视化的正确处理过程。

这些仿真程序可以测试最坏状况（极限的负载）和临界时间的程序顺序。

6、项目的维护和组织

6.1 项目的变化

"我可以用这种方法把它做的更好"

当创建可视化时，经常会出现最初采用的方案导致了程序死锁或复杂的解决方案。

应该考虑在项目重新评估阶段采取较简单的方案并做出改动。在项目完成之前，不应该执行这些改动。一旦当前阶段完成后才会做出这些改动。

项目改动应有文档，先前的项目状态必须存档而且在需要时必须通知客户。

备注:

对项目所做的任何改变，无论是基于客户要求所做的修改还是程序员执行变化和错误修所做的修改，都必须记录。这有助于以后回顾每个项目步骤的书面记录。



备忘录记录的格式和方法应该与客户商定。

6.1.1 清理项目



当程序做大量改动或集成可视化的过程中，都将产生大量的不再需要的数据。这些数据可以从工程中删除。

- 仿真或测试页：这些页面可以保留在项目中以便以后测试，然而，用户不能导航这些测试页面（页面更改）。
- 不需要的位图。

项目的维护和组织

6.2 软件分类和存储

在创建可视化时，如果项目做出较大的改变，那么必须将当前的项目状态存档（备份拷贝）。



不要把备份的拷贝存储在开发的PC上。备份应该总是保存在外部的服务器上（备份）或适当的存储介质上。

备注:

为了避免兼容性问题，所有与可视化一起使用和测试的从属的软件包都应该与当前的项目状态一起存档（通信驱动器，控制软件，外部服务工具等）。

在存档时，用户可以使用带有版本校验的不同的数据库。

6.3 把项目让渡给客户



项目的当前状态应该作为一个完整的设置或一个预先安装的存储媒介 ((CF卡等) 让渡给客户。应该避免拷贝和拆散单个文件。

这个设置应该包含用于集成的所有软件包，也包括安装向导。尤其是在软件包的安装顺序包含从属时尤为必要。

项目的维护和组织

6.4 文档

文件有几种不同的类型:

- 工程文件
- 用户文件
- 可可视化的在线帮助



"好的软件设计说明书是一个好文档的基础"

6.4.1 项目文档

在创建项目时，应该为开发的每个阶段做记录文档。项目文档用于记录所有的扩展和变化。

备注:

从项目分析到项目完成的文档中，程序员和客户对机器元素使用相同的说明和名称是十分重要的。

6.4.2 用户文档

用户文档为用户提供了操作机器的支持。

用户文档必须在不同的操作人员、维护技术员和其他操作可视化的人员之间进行区分。普通的操作者不需要任何有关启动、维护的信息以及类似的问题。

此外，用户文档应该为必要的操作提供每页的屏幕照片。

6.4.3 在线帮助

在线帮助可以看作自己的项目。帮助文件用于补充用户文档。在计划工作量时必须考虑在线帮助的创建。

在Windows中，可以使用不同的方式来创建和调用在线帮助。

如果有需要，可视化程序可以支持上下文敏感的帮助。

小结

7、小结

在结束完这一培训模块后，可视化尚未结束。

关于创建可视化系统项目，可以写成几本书。

这个培训模块为执行可视化应用提供了一个向导。

ELECTRONIC DOCUMENT

Notes

ELECTRONIC DOCUMENT

小结

培训模块综述

- | | |
|-------------------------------------|-------------------------------|
| TM200 – 贝加莱B&R 公司介绍** | TM600 – 图文显示的基础 |
| TM201 – 贝加莱B&R 产品系列** | TM601 – 贝加莱人机界面产品** |
| TM210 – Automation Studio™ 基础 | TM610 – ASiV 的基础 |
| TM211 – Automation Studio™ 在线通信 | TM620 – ASiV 的维护* |
| TM212 – 自动化对象 (Target) ** | TM630 – 图文显示的编程规则 |
| TM213 – 自动化运行 (Runtime) 系统 | TM640 – ASiV 报警系统 |
| TM220 – 维护信息* | TM650 – ASiV 的国际化操作 |
| TM221 – 自动化组件和出错信息查询* | TM660 – ASiV 的远程操作 |
| TM223 – Automation Studio™ 诊断 | TM670 – ASiV 高级应用 |
| TM230 – 结构化软件编程 | TM700 – Automation Net PVI |
| TM231 – 面向机器设备的Automation Studio™ * | TM701 – PVI 通信* |
| TM240 – 梯形图(LAD) | TM710 – PVI DLL 编程 |
| TM241 – 功能块图 (FBD)* | TM711 – PVI 的服务 |
| TM242 – 连续功能图 (CFC)* | TM712 – PVIControl.NET |
| TM243 – 顺序功能图 (SFC)* | TM720 – PVI 维护和诊断* |
| TM245 – 指令表 (IL)* | TM730 – PVI OPC |
| TM246 – 结构文本 (ST) | TM800 – APROL 系统概念 |
| TM247 – Automation Basic (AB)* | TM801 – APROL 工程设计基础 |
| TM248 – ANSI C | TM810 – APROL 安装, 配置和恢复* |
| TM250 – 内存管理和数据存储 | TM811 – APROL 运行(Runtime) 系统* |
| TM260 – Automation Studio™ 函数库I | TM812 – APROL 操作员管理 |
| TM261 – Automation Studio™ 函数库 II* | TM813 – APROL XML 查询* |
| TM264 – 定时处理单元 (TPU) * | TM814 – APROL 审计追踪* |
| TM400 – 运动控制的基础 | TM820 – APROL 维护* |
| TM401 – 贝加莱B&R 运动控制产品** | TM830 – APROL 项目工程设计 |
| TM402 – 运动控制系统的计算* | TM840 – APROL 参数管理和配方 |
| TM410 – ASiM 的基础 | TM850 – APROL 控制器配置和INA 通讯 |
| TM440 – ASiM的基本功能 | TM860 – APROL 库设计 |
| TM441 – ASiM多轴运动功能 | TM861 – APROL 通讯互联* |
| TM445 – ACOPOS ACP10 软件 | TM865 – APROL 库指导手册 |
| TM446 – 电子凸轮* | TM870 – APROL Python 编程* |
| TM447 – ACOPOS 智能过程技术 (SPT) * | TM880 – APROL 报表* |
| TM450 – ACOPOS 控制理念和控制器设置 | ** 查看产品目录 |
| TM460 – 启动B&R 电机* | * 即将出版 |

全球总部

Bernecker+Rainer Industrie-Elektronik Ges.m.b.H.

B&R Straße 1

A-5142 Eggelsberg 奥地利

Tel.: +43(0)7748/6586-0

Fax: +43(0)7748/6586-26

info@br-automation.com

www.br-automation.com

中国总部

贝加莱工业自动化（上海）有限公司

上海市漕宝路70号光大会展中心C座16楼

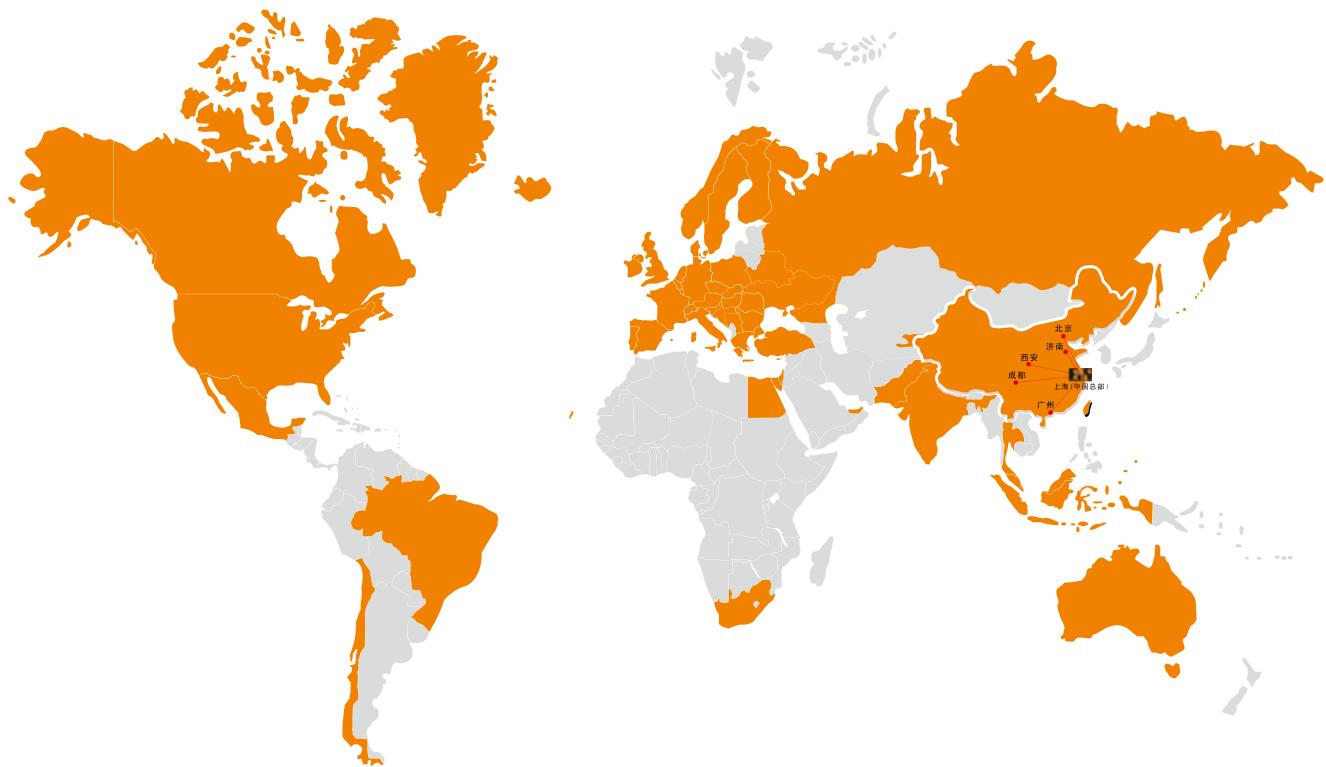
Tel.: +86/(0)21/6432 6000

Fax: +86/(0)21/6432 6108

info.cn@br-automation.com

www.br-automation.cn

全球50多个国家超过120个分支机构 www.br-automation.com/contact



中国总部



中国办事处

Austria · Australia · Belgium · Belarus · Brazil · Bulgaria · Canada · Chile · China · Croatia · Cyprus · Czech Republic · Denmark · Egypt · Emirates · Finland · France · Germany · Greece · Hungary · India · Indonesia · Ireland · Israel · Italy · Korea · Kyrgyzstan · Malaysia · Mexico · The Netherlands · Norway · Pakistan · Poland · Portugal · Romania · Russia · Singapore · Slovakia · Slovenia · South Africa · Spain · Sweden · Switzerland · Thailand · Turkey · Ukraine · United Kingdom · USA