策略风险定位方法、装置、电子设备及存储介质

相关申请的交叉引用

本申请要求于2021年12月21日提交的，申请名称为“策略风险定位方法、装置、电子设备及存储介质”的、中国专利申请号为“202111575104.1”的优先权，该中国专利申请的全部内容通过引用结合在本申请中。

技术领域

本申请涉及金融领域中的量化组件，具体涉及一种策略风险定位方法、装置、电子设备及存储介质。

背景技术

在金融领域中，量化的学习以及操作均具有一定门槛，从而导致普通用户在量化过程中所构建的策略容易出现风险，致使策略常常无法正常运行，从而使得量化操作失败。因此亟需提出一种方案以克服上述缺陷。

技术解决方案

本申请的一个目的在于提出一种策略风险定位方法、装置、电子设备及存储介质，能够提高所构建策略的安全性。

根据本申请实施例的一方面，公开了一种策略风险定位方法，所述方法包括：

响应于在画布区域对于控件的拖拽操作，构建待运行的策略；

响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果；

响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果；

响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险。

根据本申请实施例的一方面，公开了一种策略风险定位装置，所述装置包括：

策略构建模块，配置为响应于在画布区域对于控件的拖拽操作，构建待运行的策略；

全局检测模块，配置为响应于第一策略检测条件的触发，对所述画布

区域中所有控件的全局组成进行检测，得到全局检测结果；

控件内检测模块，配置为响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果；

控件间检测模块，配置为响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果；

风险定位模块，配置为基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，遍历各条件控件的内部函数；

基于所述内部函数的函数名称以及参数名称，获取所述画布区域中所有条件控件所使用的行情标的的数量，得到描述有行情标的数量的全局检测结果；

若所述全局检测结果所描述的行情标的数量大于预设的标的数量阈值，则确定所述策略在行情标的数量处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，对与所述策略的开始框相隔离的孤岛控件进行检测，得到描述有孤岛控件的全局检测结果；

基于所述全局检测结果所描述的孤岛控件，定位所述孤岛控件，并确定所述策略在所述孤岛控件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

沿开始框所在路径向下逐级遍历控件，得到遍历到的控件集合；

将所述遍历到的控件集合从所述画布区域中所有控件剔除，将剩余的控件作为所述孤岛控件。

在本申请的一示例性实施例中，所述装置配置为：

沿叶子节点控件所在路径向上逐级遍历控件，直到遍历到位于所述叶子节点控件所在路径的顶点的顶点节点控件；

若所述顶点节点控件不是开始框控件，则将沿所述叶子节点控件所在路径遍历到的各控件作为所述孤岛控件。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，对各叶子节点控件所属节点类型进行检测，得到描述有属于非事件框的叶子节点控件的全局检测结果；

基于所述全局检测结果所描述的属于非事件框的叶子节点控件，定位属于非事件框的叶子节点控件，并确定所述策略在属于非事件框的叶子节点控件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第二策略检测条件的触发，遍历各控件的内部参数；基于所述内部参数使用到的全局变量，确定控件所遗漏的全局变量，得到描述有控件所遗漏的全局变量的控件内检测结果；

基于所述控件内检测结果所描述的控件所遗漏的全局变量，确定所述

策略在所述控件所遗漏的全局变量处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第二策略检测条件的触发，基于检测到的各控件的内部参数，确定所述策略中的组合条件的逻辑合法性，得到描述有所述组合条件的逻辑合法性的控件内检测结果；

基于所述组合条件的逻辑合法性，定位所述策略中的逻辑非法的组合条件，并确定所述策略在所述逻辑非法的组合条件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第三策略检测条件的触发，对属于条件框的控件进行检测，确定条件框的出口连线的定义，得到描述有出口连线的定义的控件间检测结果；

基于所述控件间检测结果所描述的出口连线的定义，定位条件框的错误连线，并确定所述策略在所述错误连线处存在风险。

根据本申请实施例的一方面，公开了一种电子设备，包括：一个或多个处理器；存储装置，用于存储一个或多个程序，当所述一个或多个程序被所述一个或多个处理器执行时，使得所述电子设备实现以上任一项实施例。

根据本申请实施例的一方面，公开了一种计算机程序介质，其上存储有计算机可读指令，当所述计算机可读指令被计算机的处理器执行时，使计算机执行以上任一项实施例。

根据本申请实施例的一个方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述各种可选实现方式中提供的方法。

本申请所提供的策略风险定位方法，能够为用户提供可视化的控件操作途径，使得用户能够通过拖拽控件构建待运行的策略。并且针对所构建的策略，本申请通过对其进行全局检测、控件内检测以及控件间检测定位其风险所在，提高了风险定位的完善程度，从而提高了所构建策略的安全性。

本申请的其他特性和优点将通过下面的详细描述变得显然，或部分地通过本申请的实践而习得。

应当理解的是，以上的一般描述和后文的细节描述仅是示例性的，并不能限制本申请。

附图说明

通过参考附图详细描述其示例实施例，本申请的上述和其它目标、特征及优点将变得更加显而易见。

图 1 示出了根据本申请一个实施例的策略风险定位方法的流程图。

图 2 示出了根据本申请一个实施例的孤岛控件的示意图。
图 3 示出了根据本申请一个实施例的用户与量化组件的交互示意图。
图 4 示出了根据本申请一个实施例的策略风险定位装置的框图。
图 5 示出了根据本申请一个实施例的电子设备硬件图。

具体实施方式

现在将参考附图更全面地描述示例实施方式。然而，示例实施方式能够以多种形式实施，且不应被理解为限于在此阐述的范例；相反，提供这些示例实施方式使得本申请的描述将更加全面和完整，并将示例实施方式的构思全面地传达给本领域的技术人员。附图仅为本申请的示意性图解，并非一定是按比例绘制。图中相同的附图标记表示相同或类似的部分，因而将省略对它们的重复描述。

此外，所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多示例实施方式中。在下面的描述中，提供许多具体细节从而给出对本申请的示例实施方式的充分理解。然而，本领域技术人员将意识到，可以实践本申请的技术方案而省略所述特定细节中的一个或更多，或者可以采用其它的方法、组元、步骤等。在其它情况下，不详细示出或描述公知结构、方法、实现或者操作以避免喧宾夺主而使得本申请的各方面变得模糊。

附图中所示的一些方框图是功能实体，不一定必须与物理或逻辑上独立的实体相对应。可以采用软件形式来实现这些功能实体，或在一个或多个硬件模块或集成电路中实现这些功能实体，或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

本申请提供了一种策略风险定位方法，主要应用于金融领域中的量化组件。具体的，本申请中的量化组件供用户在其画布区域中对控件进行拖拽操作，结合控件之间的连线构建得到对应的策略。控件通常包括开始框、条件框以及事件框。构建完成的策略可用于进行实盘运行或者历史回测。实盘运行指的是将策略运用于真实交易场景，实时进行交易。历史回测指的是利用历史数据检测策略的有效性。

图 1 示出了本申请一实施例的策略风险定位方法的流程图，该方法的示例性执行主体为量化组件，该方法包括：

步骤 S110、响应于在画布区域对于控件的拖拽操作，构建待运行的策略；

步骤 S120、响应于第一策略检测条件的触发，对画布区域中所有控件的全局组成进行检测，得到全局检测结果；

步骤 S130、响应于第二策略检测条件的触发，对画布区域中各控件的内部参数进行检测，得到控件内检测结果；

步骤 S140、响应于第三策略检测条件的触发，对画布区域中相邻控件

的连接关系进行检测，得到控件间检测结果；

步骤S150、基于全局检测结果、控件内检测结果以及控件间检测结果，定位策略中存在的风险。

本申请实施例中，量化组件将其画布区域提供给用户，使得用户可在其中进行策略的构建。响应于在画布区域中对于控件的拖拽操作，量化组件构建对应的策略以待运行。本实施例中策略用于自动对标的进行交易处理。

在策略构建过程中，响应于第一策略检测条件的触发，量化组件对画布区域中所有控件的全局组成进行检测，得到全局检测结果。即，将画布区域中所有控件作为一个检测对象，检测该检测对象的全局组成，得到全局检测结果。

在策略构建过程中，响应于第二策略检测条件的触发，量化组件对画布区域中各控件的内部参数进行检测，得到控件内检测结果。即，将画布区域中各控件分别作为一个检测对象，分别检测各检测对象的内部参数，得到控件内检测结果。

在策略构建过程中，响应于第三策略检测条件的触发，量化组件对画布区域中相邻控件的连接关系进行检测，得到控件间检测结果。即，将画布区域中两两相邻的一组组控件分别作为一对检测对象，分别检测各对检测对象的连接关系，得到控件间检测结果。

进而基于全局检测结果、控件内检测结果以及控件间检测结果，定位策略中存在的风险。

由此可见，本申请所提供的策略风险定位方法，能够为用户提供可视化的控件操作途径，使得用户能够通过拖拽控件构建待运行的策略。并且针对所构建的策略，本申请通过对其进行全局检测、控件内检测以及控件间检测定位其风险所在，提高了风险定位的完善程度，从而提高了所构建策略的安全性。

在一实施例中，响应于第一策略检测条件的触发，对画布区域中所有控件的数量进行检测，得到描述有所有控件数量的全局检测结果。若全局检测结果所描述的所有控件数量大于预设的控件数量阈值，则确定策略在所有控件数量处存在风险。

该实施例中，从控件数量的维度进行全局检测。具体的，第一策略检测条件触发后，检测画布区域中用于构建策略的所有控件的数量。若所有控件数量大于预设的控件数量阈值，则说明用于构建策略所使用的控件过多，超出量化组件的承载负荷，因此存在风险。进一步的，在确定策略在所有控件数量处存在风险后，可以触发错误提醒，以提醒用户需要其在画布区域中构建的策略会由于所有控件数量超出限制而无法正常运行。

在一实施例中，响应于第一策略检测条件的触发，遍历各条件控件的内部函数；基于内部函数的函数名称以及参数名称，获取画布区域中所有

条件控件所使用的行情标的的数量，得到描述有行情标的数量的全局检测结果。若全局检测结果所描述的行情标的数量大于预设的标的数量阈值，则确定策略在行情标的数量处存在风险。

该实施例中，从行情标的数量的维度进行全局检测。具体的，由于条件控件的本质是在量化组件中调用行情数据，而量化组件需要对行情数据限频以保护后台，故需要对使用到的行情标的数量进行限制。第一策略检测条件触发后，检测画布区域中的所有条件控件，进而确定所有条件控件所使用的行情标的的数量。若行情标的数量大于预设的标的数量阈值，则说明用于构建策略所使用的行情标的过多，会对后台造成威胁，因此存在风险。进一步的，在确定策略在行情标的数量处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于所有行情标的数量超出限制而无法正常运行。

在一实施例中，响应于第一策略检测条件的触发，对与策略的开始框相隔离的孤岛控件进行检测，得到描述有孤岛控件的全局检测结果。基于全局检测结果所描述的孤岛控件，定位孤岛控件，并确定策略在孤岛控件处存在风险。

该实施例中，从全局拓扑的维度进行全局检测。具体的，第一策略检测条件触发后，检测画布区域中的孤岛控件。参考图 2 中虚线框所圈选的孤岛控件，孤岛控件指的是与开始框相隔离，不存在直接或间接连接关系的控件。若存在孤岛控件，则说明这些孤岛控件无法作为所构建策略的一部分而发挥作用，因此存在风险。进一步的，在确定策略在孤岛控件处存在风险后，可以触发警告提醒，以提醒用户其在画布区域中构建的策略会由于存在孤岛控件而可能出错。

在一实施例中，沿开始框所在路径向下逐级遍历控件，得到遍历到的控件集合。将遍历到的控件集合从画布区域中所有控件剔除，将剩余的控件作为孤岛控件。

该实施例中，定位画布区域中作为根节点控件的开始框，进而沿开始框所在路径向下逐级遍历控件，遍历到的控件均与开始框存在直接或间接连接关系，不属于孤岛控件。当遍历了开始框的所有路径后，遍历到的控件集合即包含了所有不属于孤岛控件的控件，并且，不在该遍历到的控件集合中的控件均属于孤岛控件。因此，将该遍历到的控件集合从画布区域中所有控件剔除，进而将剩余的控件作为孤岛控件。

在一实施例中，沿叶子节点控件所在路径向上逐级遍历控件，直到遍历到位于叶子节点控件所在路径的顶点的顶点节点控件。若顶点节点控件不是开始框，则将沿叶子节点控件所在路径遍历到的各控件作为孤岛控件。

该实施例中，定位画布区域中最底层的各叶子节点控件，进而沿各叶子节点控件所在路径向上逐级控件，直到遍历到对应的顶点节点控件。顶点节点控件指的是不存在父节点，位于所在路径顶点的控件。当存在孤岛控件时，画布区域中的顶点节点控件有多个，其中只有一个顶点节点控件

是作为根节点控件的开始框。

若某个叶子节点控件的顶点节点控件不是开始框，则说明从该叶子节点控件到其对应的顶点节点控件的所有控件均与开始框相隔离，因此沿该叶子节点控件所在路径遍历到的所有控件均属于孤岛控件。

在一实施例中，响应于第一策略检测条件的触发，对各叶子节点控件所属节点类型进行检测，得到描述有属于非事件框的叶子节点控件的全局检测结果。基于全局检测结果所描述的属于非事件框的叶子节点控件，定位属于非事件框的叶子节点控件，并确定策略在属于非事件框的叶子节点控件处存在风险。

该实施例中，从叶子节点控件类型的维度进行全局检测。具体的，为保证策略正常运行，路径的终点必须为事件框。即，叶子节点控件必须为事件框。第一策略检测条件触发后，检测画布区域中的各叶子节点控件所属节点类型，确定哪些叶子节点控件属于事件框，哪些叶子节点控件属于非事件框。其中，属于非事件框的叶子节点控件会影响策略的正常运行，存在风险。进一步的，在确定策略在属于非事件框的叶子节点控件处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于叶子节点控件不属于事件框而无法正常运行。

在一实施例中，第一策略检测条件包括：检测到对策略所作的回测操作，检测到对策略所作的实盘运行操作，检测到对策略所作的逻辑检验操作。

该实施例中，任一第一策略检测条件均能够触发全局检测。用户对策略进行回测操作会触发全局检测；用户对策略进行实盘运行操作也会触发全局检测；用户对策略进行逻辑检验操作也会触发全局检测。

在一实施例中，从策略类型以及函数类型的维度进行控件内检测。策略类型描述的是用户账户对应的账户类型，账户类型包含期货账户以及证券账户。每个控件所使用的函数都有对应的函数类型，有的函数类型仅适用于期货账户，有的函数类型仅适用于证券账户，有的函数类型两者均可适用。为了保证账户类型与函数类型的一致，对策略类型以及函数类型进行检测。

具体的，第二策略检测条件触发后，从开始框提取出策略类型，从而确定账户类型；遍历控件中使用到的函数，查看函数类型，进而与账户类型进行比较，确定二者是否相符。若不相符，则会影响策略的正常运行，存在风险。进一步的，在确定策略在策略类型以及函数类型之间的匹配上存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于策略类型与账户类型不符或者由于函数类型与账户类型不符而无法正常运行。

在一实施例中，响应于第二策略检测条件的触发，遍历各控件的内部参数；基于内部参数使用到的全局变量，确定控件所遗漏的全局变量，得

到描述有控件所遗漏的全局变量的控件内检测结果。基于控件内检测结果所描述的控件所遗漏的全局变量，确定策略在控件所遗漏的全局变量处存在风险。

该实施例中，从全局变量的维度进行控件内检测。具体的，第二策略检测条件触发后，遍历各控件的内部参数。遍历到属于开始框的控件时，根据开始框中定义的全局变量，建立全局变量集合；在遍历到其他控件时，将其他控件所用到的全局变量从全局变量集合中剔除。所有控件遍历完毕后，全局变量集合中剩余的全局变量属于没有使用到的全局变量，即，控件所遗漏的全局变量。若存在控件所遗漏的全局变量，可能会存在操作风险。进一步的，在确定策略在控件所遗漏的全局变量处存在风险后，可以触发警告提醒，以提醒用户其在画布区域中构建的策略会由于遗漏有全局变量而可能出错。

在一实施例中，从必填参数的维度进行控件内检测。必须填写的参数为必填参数，若漏填必填参数，会影响策略的正常运行。具体的，预先配置一张必填参数的映射表。第二策略检测条件触发后，遍历各控件的内部参数，检测参数是否为手动输入且值为空，若是且位于映射表中，则说明参数为漏填的必填参数，存在风险。进一步的，在确定策略在漏填的必填参数处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于漏填有必填参数而无法正常运行。

在一实施例中，响应于第二策略检测条件的触发，基于检测到的各控件的内部参数，确定策略中的组合条件的逻辑合法性，得到描述有组合条件的逻辑合法性的控件内检测结果。基于组合条件的逻辑合法性，定位策略中的逻辑非法的组合条件，并确定策略在逻辑非法的组合条件处存在风险。

该实施例中，从组合条件的逻辑合法维度进行控件内检测。具体的，第二策略检测条件触发后，检测各控件的内部参数，进而从中提取出控件所使用到的子条件。进而基于子条件之间的逻辑关系，检测由多个子条件组合而成的组合条件是否逻辑合法。若组合条件逻辑不完整，则会导致策略运行出错。进一步的，在确定策略在逻辑非法的组合条件处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于组合条件逻辑非法而无法正常运行。

其中，每一个子条件，逻辑符号以及括号都在组合条件中使用名为QCGAN（Quant Condition Group AST Node）的数据结构包裹，QCAGN 里面使用枚举值来标识其所属类型。类型包括：条件，逻辑符号，左括号，以及右括号。QCAGN 的输入顺序是固定的，故采用数组进行存储，另外使用一张表来维护 QCAGN 的下表位置和其对应的错误信息。每次判断到非法的 QCAGN，便将其在数组的下标和对应的错误信息记录到表里面，另外使用栈来做左右括号闭合的判断：每次遇到左括号，便将其数组下标压栈；每次遇到右括号便弹出一个。针对 QCAGN 所作处理，每次会将本次处理

的 QCAGN 和前一个 QCAGN 的类型进行判断，以确定组合条件是否逻辑合法。判断如下：

1、针对条件类型的 QCAGN，若其不是出现在开始位置，并且其前一个 QCAGN 的类型不是左括号或者逻辑符号，则其逻辑非法；其他情况下逻辑合法。

2、针对逻辑符号类型的 QCAGN，若其前一个 QCAGN 的类型不是右括号或者条件，则其逻辑非法；其他情况下逻辑合法。

3、针对左括号类型的 QCAGN，若其不是出现在开始位置，并且其前一个 QCAGN 的类型不是逻辑符号或者左括号，则其逻辑非法；其他情况下逻辑合法。

4、针对右括号类型的 QCAGN，先判断栈是否空的，若栈是空的代表没有对应的左括号，进而再判断其前一个 QCAGN 的类型。若其前一个 QCAGN 的类型不是右括号或者条件，则其逻辑非法；其他情况下逻辑合法。

在一实施例中，从组合条件的子条件维度进行控件内检测。子条件是构成组合条件的重要部分，用户如果创建子条件却不使用，会存在操作失误的风险。具体的，在组合条件的数据结构中维护一个用于记录子条件使用记录的表。该表对使用到的子条件进行计数。第二策略检测条件触发后，查阅该表定位计数为 0 的子条件，并确定策略在计数为 0 的子条件处存在风险。进一步的，在确定策略在计数为 0 的子条件处存在风险后，可以触发警告提醒，以提醒用户其在画布区域中构建的策略会由于组合条件中存在未使用到的子条件而可能出错。

在一实施例中，从信号条件所适用的标的维度进行控件内检测。信号条件本质是通过量化组件调用行情数据，但这些行情数据是针对特定标的的。例如对于股票标的不存在希腊值数据，希腊值数据只针对期权标的。不适用于信号条件的标的会影响策略的正常运行。具体的，对函数对应的信号条件进行封装，得到用于维护函数信息的函数表。该函数表中维护的函数信息包括函数适用的账户类型以及函数适用的标的类型。第二策略检测条件触发后，查询信号条件的标的信息，将查询到的标的信息于函数表中维护的函数信息进行对比，从而确定信号条件所适用的标的是否存在风险。进一步的，在确定策略在信号条件所适用的标的处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于信号条件所使用的标的存在错误而无法正常运行。

在一实施例中，从账户条件所适用的标的维度进行控件内检测。账户条件的本质是通过量化组件调用账户相关数据，但是账户也是针对特定标的的。例如股票账户无法调出期货标的的数据。不适用于账户条件的标的会影响策略的正常运行。具体的，第二策略检测条件触发后，从开始框提取出策略类型；遍历控件收集策略中的标的，与策略类型进行对比，从而确定账户条件所适用的标的是否存在风险。进一步的，在确定策略在账户条件所适用的标的处存在风险后，可以触发错误提醒，以提醒用户其在画

布区域中构建的策略会由于账户条件所使用的标的存在错误而无法正常运行。

在一实施例中，第二策略检测条件包括：检测到对控件所作的编辑操作。该编辑操作包括编辑属性栏，编辑组合条件窗口。

在一实施例中，响应于第三策略检测条件的触发，对属于条件框的控件进行检测，确定条件框的出口连线的定义，得到描述有出口连线的定义的控件间检测结果。基于控件间检测结果所描述的出口连线的定义，定位条件框的错误连线，并确定策略在错误连线处存在风险。

该实施例中，从条件框的出口一般最多可以拉出两条连线，用户可以编辑这两条连线的定义（“是”，“否”）。如果这两条连线定义相同，则会影响策略的正常运行。具体的，第二策略检测条件触发后，遍历控件，当遍历到条件框时，判断出口的连线的定义是否相同，从而确定是否存在错误连线。进一步的，在确定策略在错误连线处存在风险后，可以触发错误提醒，以提醒用户其在画布区域中构建的策略会由于错误连线而无法正常运行。

在一实施例中，第三策略检测条件包括：检测到对控件所作的移除操作，检测到对控件所作的连线操作。

图 3 示出了本申请一实施例的用户与量化组件的交互示意图。

该实施例中，对量化组件中的画布区域进行检测是通向策略回测以及实盘运行的必经过程。若用户没有点击“逻辑检验”按钮，直接点击“策略回测”按钮或者“实盘运行”按钮，量化组件的系统也会先对画布区域中构建的策略进行检测，检测通过后再执行策略回测或者实盘运行。

具体的，用户编辑画布区域以构建策略，完成编辑后，点击“逻辑检验”按钮或是点击策略回测”按钮或是点击“实盘运行”按钮均会触发对策略进行检测。

若策略通过检测，系统前端提醒通过，进而系统进入下一阶段；同时，由于策略逻辑通过检测意味着不存在影响策略正常运行的错误，若检测出策略存在可能导致出错的风险，可选的，系统前端还会进行警告提醒。

若策略未通过检测，系统前端进行错误提醒（例如通过设定颜色显示控件、或者在两个控件之间的连线上显示错误标记“×”），用户再次编辑画布区域以去除错误。在用户编辑画布区域的过程中，可选的，系统可以同步检测策略是否通过，并根据检测结果作出相应的提醒。

进一步的，本实施例中可以根据全局检测结果、控件内检测结果、控件间检测结果的情况来衡量本次量化交易策略的异常参数，并在异常参数大于或者等于预设的异常阈值时，对本次量化交易策略进行重建，或者提醒用户本次量化策略异常较多是否重建；在异常参数小于异常阈值时，对本次量化交易策略中的异常和错误进行修改和完善。

具体的，在计算异常参数时，获取全局检测结果、控件内检测结果、控件间检测结果中发生异常的异常数量，分别对应为第一异常数量*Ece*_1、第二异常数量*Ece*_2、第三异常数量*Ece*_3，同时获取全局检测结果、控件内检测结果、控件间检测中包括的策略运行的路径总数，分别对应为第一路径数量*Path*_1、第二路径数量*Path*_2以及第三路径数量*Path*_3。通过如下公式计算异常参数*Para*_*ab*：

$$Para_ab=\alpha\cdot\frac{Ece_1}{Path_1}+\beta\cdot\frac{Ece_2}{Path_2}+\gamma\cdot\frac{Ece_3}{Path_3}$$

其中，α、β、γ表示各检测方式对应的异常因子。

通过上述方式，在异常较为严重的情况下直接建议用户重建策略，避免了修改大量的异常问题加大的时间成本，提高了量化交易策略编写和运行的效率。

图4示出了根据本申请一实施例的策略风险定位装置，所述装置包括：

策略构建模块210，配置为响应于在画布区域对于控件的拖拽操作，构建待运行的策略；

全局检测模块220，配置为响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果；

控件内检测模块230，配置为响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果；

控件间检测模块240，配置为响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果；

风险定位模块250，配置为基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，遍历各条件控件的内部函数；

基于所述内部函数的函数名称以及参数名称，获取所述画布区域中所有条件控件所使用的行情标的的数量，得到描述有行情标的数量的全局检测结果；

若所述全局检测结果所描述的行情标的数量大于预设的标的数量阈值，则确定所述策略在行情标的数量处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，对与所述策略的开始框相隔离的孤岛控件进行检测，得到描述有孤岛控件的全局检测结果；

基于所述全局检测结果所描述的孤岛控件，定位所述孤岛控件，并确定所述策略在所述孤岛控件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

沿开始框所在路径向下逐级遍历控件，得到遍历到的控件集合；

将所述遍历到的控件集合从所述画布区域中所有控件剔除，将剩余的

控件作为所述孤岛控件。

在本申请的一示例性实施例中，所述装置配置为：

沿叶子节点控件所在路径向上逐级遍历控件，直到遍历到位于所述叶子节点控件所在路径的顶点的顶点节点控件；

若所述顶点节点控件不是开始框控件，则将沿所述叶子节点控件所在路径遍历到的各控件作为所述孤岛控件。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第一策略检测条件的触发，对各叶子节点控件所属节点类型进行检测，得到描述有属于非事件框的叶子节点控件的全局检测结果；

基于所述全局检测结果所描述的属于非事件框的叶子节点控件，定位属于非事件框的叶子节点控件，并确定所述策略在属于非事件框的叶子节点控件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第二策略检测条件的触发，遍历各控件的内部参数；基于所述内部参数使用到的全局变量，确定控件所遗漏的全局变量，得到描述有控件所遗漏的全局变量的控件内检测结果；

基于所述控件内检测结果所描述的控件所遗漏的全局变量，确定所述策略在所述控件所遗漏的全局变量处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第二策略检测条件的触发，基于检测到的各控件的内部参数，确定所述策略中的组合条件的逻辑合法性，得到描述有所述组合条件的逻辑合法性的控件内检测结果；

基于所述组合条件的逻辑合法性，定位所述策略中的逻辑非法的组合条件，并确定所述策略在所述逻辑非法的组合条件处存在风险。

在本申请的一示例性实施例中，所述装置配置为：

响应于所述第三策略检测条件的触发，对属于条件框的控件进行检测，确定条件框的出口连线的定义，得到描述有出口连线的定义的控件间检测结果；

基于所述控件间检测结果所描述的出口连线的定义，定位条件框的错误连线，并确定所述策略在所述错误连线处存在风险。

下面参考图 5 来描述根据本申请实施例的电子设备 30。图 5 显示的电子设备 30 仅仅是一个示例，不应对本申请实施例的功能和使用范围带来任何限制。

如图 5 所示，电子设备 30 以通用计算设备的形式表现。电子设备 30 的组件可以包括但不限于：上述至少一个处理单元 310、上述至少一个存储单元 320、连接不同系统组件（包括存储单元 320 和处理单元 310）的总线 330。

其中，所述存储单元存储有程序代码，所述程序代码可以被所述处理

单元 310 执行，使得所述处理单元 310 执行本说明书上述示例性方法的描述部分中描述的根据本申请各种示例性实施方式的步骤。例如，所述处理单元 310 可以执行如图 1 中所示的各个步骤。

存储单元 320 可以包括易失性存储单元形式的可读介质，例如随机存取存储单元（RAM）3201 和/或高速缓存存储单元 3202，还可以进一步包括只读存储单元（ROM）3203。

存储单元 320 还可以包括具有一组（至少一个）程序模块 3205 的程序/实用工具 3204，这样的程序模块 3205 包括但不限于：操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。

总线 330 可以为表示几类总线结构中的一种或多种，包括存储单元总线或者存储单元控制器、外围总线、图形加速端口、处理单元或者使用多种总线结构中的任意总线结构的局域总线。

电子设备 30 也可以与一个或多个外部设备 400（例如键盘、指向设备、蓝牙设备等）通信，还可与一个或者多个使得用户能与该电子设备 30 交互的设备通信，和/或与使得该电子设备 30 能与一个或多个其它计算设备进行通信的任何设备（例如路由器、调制解调器等等）通信。这种通信可以通过输入/输出（I/O）接口 350 进行。输入/输出（I/O）接口 350 与显示单元 340 相连。并且，电子设备 30 还可以通过网络适配器 360 与一个或者多个网络（例如局域网（LAN），广域网（WAN）和/或公共网络，例如因特网）通信。如图所示，网络适配器 360 通过总线 330 与电子设备 30 的其它模块通信。应当明白，尽管图中未示出，可以结合电子设备 30 使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID 系统、磁带驱动器以及数据备份存储系统等。

通过以上的实施方式的描述，本领域的技术人员易于理解，这里描述的示例实施方式可以通过软件实现，也可以通过软件结合必要的硬件的方式来实现。因此，根据本申请实施方式的技术方案可以以软件产品的形式体现出来，该软件产品可以存储在一个非易失性存储介质（可以是 CD-ROM，U 盘，移动硬盘等）中或网络上，包括若干指令以使得一台计算设备（可以是个人计算机、服务器、终端装置、或者网络设备等）执行根据本申请实施方式的方法。

在本申请的示例性实施例中，还提供了一种计算机可读存储介质，其上存储有计算机可读指令，当所述计算机可读指令被计算机的处理器执行时，使计算机执行上述方法实施例部分描述的方法。

根据本申请的一个实施例，还提供了一种用于实现上述方法实施例中的方法的程序产品，其可以采用便携式紧凑盘只读存储器(CD-ROM)并包括程序代码，并可以在终端设备，例如个人电脑上运行。然而，本申请的程序产品不限于此，在本文件中，可读存储介质可以是任何包含或存储程序

的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

所述程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。可读存储介质的更具体的例子（非穷举的列表）包括：具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦式可编程只读存储器（EPROM 或闪存）、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了可读程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。可读信号介质还可以是可读存储介质以外的任何可读介质，该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、有线、光缆、RF 等等，或者上述的任意合适的组合。

可以以一种或多种程序设计语言的任意组合来编写用于执行本申请操作的程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如JAVA、C++等，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中，远程计算设备可以通过任意种类的网络，包括局域网（LAN）或广域网（WAN），连接到用户计算设备，或者，可以连接到外部计算设备（例如利用因特网服务提供商来通过因特网连接）。

应当注意，尽管在上文详细描述中提及了用于动作执行的设备的若干模块或者单元，但是这种划分并非强制性的。实际上，根据本申请的实施方式，上文描述的两个或更多模块或者单元的特征和功能可以在一个模块或者单元中具体化。反之，上文描述的一个模块或者单元的特征和功能可以进一步划分为由多个模块或者单元来具体化。

此外，尽管在附图中以特定顺序描述了本申请中方法的各个步骤，但是，这并非要求或者暗示必须按照该特定顺序来执行这些步骤，或是必须执行全部所示的步骤才能实现期望的结果。附加的或备选的，可以省略某些步骤，将多个步骤合并为一个步骤执行，以及/或者将一个步骤分解为多个步骤执行等。

通过以上的实施方式的描述，本领域的技术人员易于理解，这里描述的示例实施方式可以通过软件实现，也可以通过软件结合必要的硬件的方

式来实现。因此，根据本申请实施方式的技术方案可以以软件产品的形式体现出来，该软件产品可以存储在一个非易失性存储介质（可以是CD-ROM，U 盘，移动硬盘等）中或网络上，包括若干指令以使得一台计算设备（可以是个人计算机、服务器、移动终端、或者网络设备等）执行根据本申请实施方式的方法。

本领域技术人员在考虑说明书及实践这里公开的申请后，将容易想到本申请的其它实施方案。本申请旨在涵盖本申请的任何变型、用途或者适应性变化，这些变型、用途或者适应性变化遵循本申请的一般性原理并包括本申请未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本申请的真正范围和精神由所附的权利要求指出。

权 利 要 求 书

1、一种策略风险定位方法，所述方法包括：

响应于在画布区域对于控件的拖拽操作，构建待运行的策略，所述策略用于自动对标的进行交易处理；

响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果；

响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果；

响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险。

2、根据权利要求 1 所述的方法，其中，响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果，包括：响应于所述第一策略检测条件的触发，遍历各条件控件的内部函数；基于所述内部函数的函数名称以及参数名称，获取所述画布区域中所有条件控件所使用的行情标的的数量，得到描述有行情标的数量的全局检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：若所述全局检测结果所描述的行情标的数量大于预设的标的数量阈值，则确定所述策略在行情标的数量处存在风险。

3、根据权利要求 1 所述的方法，其中，响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果，包括：响应于所述第一策略检测条件的触发，对与所述策略的开始框相隔离的孤岛控件进行检测，得到描述有孤岛控件的全局检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：基于所述全局检测结果所描述的孤岛控件，定位所述孤岛控件，并确定所述策略在所述孤岛控件处存在风险。

4、根据权利要求 3 所述的方法，其中，对与所述策略的开始框相隔离的孤岛控件进行检测，包括：

沿开始框所在路径向下逐级遍历控件，得到遍历到的控件集合；

将所述遍历到的控件集合从所述画布区域中所有控件剔除，将剩余的控件作为所述孤岛控件。

5、根据权利要求 3 所述的方法，其中，对与所述策略的开始框相隔离的孤岛控件进行检测，包括：

沿叶子节点控件所在路径向上逐级遍历控件，直到遍历到位于所述叶子节点控件所在路径的顶点的顶点节点控件；

若所述顶点节点控件不是开始框控件，则将沿所述叶子节点控件所在路径遍历到的各控件作为所述孤岛控件。

6、根据权利要求 1 所述的方法，其中，响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果，包括：响应于所述第一策略检测条件的触发，对各叶子节点控件所属节点类型进行检测，得到描述有属于非事件框的叶子节点控件的全局检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：基于所述全局检测结果所描述的属于非事件框的叶子节点控件，定位属于非事件框的叶子节点控件，并确定所述策略在属于非事件框的叶子节点控件处存在风险。

7、根据权利要求 1 所述的方法，其中，所述第一策略检测条件包括：检测到对所述策略所作的回测操作，检测到对所述策略所作的实盘运行操作，检测到对所述策略所作的逻辑检验操作。

8、根据权利要求 1 所述的方法，其中，响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果，包括：响应于所述第二策略检测条件的触发，遍历各控件的内部参数；基于所述内部参数使用到的全局变量，确定控件所遗漏的全局变量，得到描述有控件所遗漏的全局变量的控件内检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：基于所述控件内检测结果所描述的控件所遗漏的全局变量，确定所述策略在所述控件所遗漏的全局变量处存在风险。

9、根据权利要求 1 所述的方法，其中，响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果，包括：响应于所述第二策略检测条件的触发，基于检测到的各控件的内部参数，确定所述策略中的组合条件的逻辑合法性，得到描述有所述组合条件的逻辑合法性的控件内检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：基于所述组合条件的逻辑合法性，定位所述策略中的逻辑非法的组合条件，并确定所述策略在所述逻辑非法的组合条件处存在风险。

10、根据权利要求 1 所述的方法，其中，所述第二策略检测条件包括：检测到对控件所作的编辑操作。

11、根据权利要求 1 所述的方法，其中，响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果，包括：响应于所述第三策略检测条件的触发，对属于条件框的控件进行检测，确定条件框的出口连线的定义，得到描述有出口连线的定义的

控件间检测结果；

基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险，包括：基于所述控件间检测结果所描述的出口连线的定义，定位条件框的错误连线，并确定所述策略在所述错误连线处存在风险。

12、根据权利要求 1 所述的方法，其中，所述第三策略检测条件包括：检测到对控件所作的移除操作，检测到对控件所作的连线操作。

13、一种策略风险定位装置，所述装置包括：

策略构建模块，配置为响应于在画布区域对于控件的拖拽操作，构建待运行的策略；

全局检测模块，配置为响应于第一策略检测条件的触发，对所述画布区域中所有控件的全局组成进行检测，得到全局检测结果；

控件内检测模块，配置为响应于第二策略检测条件的触发，对所述画布区域中各控件的内部参数进行检测，得到控件内检测结果；

控件间检测模块，配置为响应于第三策略检测条件的触发，对所述画布区域中相邻控件的连接关系进行检测，得到控件间检测结果；

风险定位模块，配置为基于所述全局检测结果、所述控件内检测结果以及所述控件间检测结果，定位所述策略中存在的风险。

14、一种电子设备，包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，当所述一个或多个程序被所述一个或多个处理器执行时，使得所述电子设备实现如权利要求 1 至 12 中任一项所述的方法。

15、一种计算机可读存储介质，其上存储有计算机可读指令，当所述计算机可读指令被计算机的处理器执行时，使计算机执行权利要求 1 至 12 中任一项所述的方法。