

К ВОПРОСУ О СОЗДАНИИ МОДЕЛЕЙ САМООБУЧЕНИЯ ПРОГРАММ ПРИ ЧЕЛОВЕКО-МАШИННОМ ВЗАИМОДЕЙСТВИИ

Д.В. Блувштейн

Управления Федерального казначейства по Костромской области, г. Кострома, Россия

Рассмотрено исследование процесса создания моделей самообучения программ при человеко-машинном взаимодействии. Одной из целей организации процесса самообучения программ, задействованных в интерфейсе, является увеличение скорости работы пользователей. Производительность пользователя находится в прямой зависимости от длительности выполнения им работы. Факторы, влияющие на работу пользователя: длительности восприятия исходной информации; длительности интеллектуальной работы; длительности физических действий пользователя; длительности реакции системы. Процесс размышления занимает почти все время, в течение которого пользователь работает с компьютером. Увеличение скорости этих размышлений приводит к увеличению скорости работы пользователя. Для реализации целей необходимо создать ряд компонент: программу самообучения; программу слежения за действиями пользователя; модули реализации дополнительных действий.

Кратко изложены принципы работы данных компонент. Программа самообучения в результате взаимодействия с программой слежения за действиями пользователя обучается различным дополнительным действиям, которые составляют правила и хранятся в базе данных. Приведен пример типичной последовательности действий пользователя при создании нового документа и работа правил.

Рассмотрены подходы, позволяющие с помощью программы самообучения для автоматизированной системы реализовать одну цель – увеличение скорости работы пользователей.

Ключевые слова: пользовательский интерфейс, взаимодействие, пользователь, самообучение, модель, диалог, алгоритм, правила.

AS FOR DEVELOPMENT OF MODELS OF SELF-STUDY PROGRAMS WITHIN HUMAN-COMPUTER INTERACTION

D.V. Bluvshstein

Department of the Federal Treasury for the Kostroma region, Kostroma, Russia

Research of the process of creating of models of self-study programs within Human-Computer Interaction has been considered. One of the goals of organizing of self-study programs process involved in the interface is to increase the speed of users' work. User's productivity is directly dependent on the duration of their work. Factors affecting their work are as following: duration of perception of the original information; duration of intellectual work; duration of the user's physical actions; duration of the reaction system. Process of thinking takes almost all the time during which user works with a computer. Increasing the speed of these reflections leads to an increase in the user's speed. To achieve the goals, it is necessary to create a number of components such as following: self-study program; tracking program for user's actions; modules for the implementation of additional actions.

Principles of functioning of these components have been briefly stated. Self-study program can be trained and perform various additional actions that make up the rules, which are kept in the database as a result of interaction with the tracking program for user's actions. An example of a typical sequence of user's actions when creating a new document and performance of the rules is given.

Approaches that let to realize one goal that is an increase in the speed of user's work using self-study program for an automated system are considered.

Keywords: user interface, interaction, user, self-study, model, dialogue, algorithm, rules.

Введение

Анализ основных понятий, необходимых для описания процесса человеко-машинного взаимодействия

Взаимодействие – обмен действиями и реакциями на эти действия между компьютером и пользователем. Пользовательский интерфейс – способ взаимодействия между человеком и компьютером.

Основными принципами создания интерфейса являются: естественность, непротиворечивость, избыточность, непосредственный доступ к системе помощи, гибкость.

Как правило, все работы по ускорению действия пользователя сводились к быстроте его работы с графическим интерфейсом при помощи клавиатуры и мыши. Распространение получило контекстное меню, диалоговые окна, открывающиеся в удобном для пользователя месте экрана. Существуют другие спо-

собы ускорения работы пользователя, например, в драйверах для мыши использовать возможность автоматически перемещать курсор к кнопкам, выбранным по умолчанию или использование «горячих клавиш» и т. п. Однако это увеличивает так же вероятность человеческих ошибок. Этот путь практически исчерпал себя.

Существует другой путь, при котором программа сама в ряде случаев может принимать решение за пользователя, задействуя модули самообучения. При решении конкретных задач пользователем с помощью приложения программа пользовательского интерфейса должна подсказывать, предлагать пользователю, какие использовать визуальные элементы управления. Для реализации этих целей нужно создать следующие компоненты: программу самообучения; программу слежения за действиями пользователя; ряд вспомогательных модулей.

При человеко-машинном взаимодействии возможны различные цели для организации процесса самообучения программ, участвующих в интерфейсе.

Исходя из основных критериев эргономики, основными целями организации процесса самообучения программ, задействованных в интерфейсе, являются: увеличение скорости работы пользователей; уменьшение количества человеческих ошибок; увеличение скорости обучения; достижение максимума удовлетворенности пользователя.

Каждую цель можно реализовать различными способами. Рассмотрим первую цель и построим модель самообучения программ для достижения ее.

Производительность пользователя находится в прямой зависимости от длительности выполнения им работы. Длительность выполнения работы пользователем зависит от ряда факторов: длительности восприятия исходной информации; длительности интеллектуальной работы; длительности физических действий пользователя; длительности реакции системы.

Взаимодействие пользователя с системой состоит из семи шагов: формирование цели действий; определение общей направленности действий; определение конкретных действий; выполнение действий; восприятие нового состояния системы; интерпретация состояния системы; оценка результата.

Из этого списка видно, что процесс размышления занимает почти все время, в течение которого пользователь работает с компьютером. Соответственно, повышение скорости этих размышлений приводит к улучшению скорости работы пользователя.

Существенно повысить скорость мышления пользователей невозможно. Можно уменьшить влияние факторов, усложняющих процесс мышления.

1. Материалы и методы

Методики по уменьшению влияния факторов, усложняющих процесс мышления, следующие: непосредственное манипулирование (НП); компенсация потери фокуса (КПФ); ограничение принятия решений (ОПР); закон Хика (ЗХ).

Можно выделить определенные этапы, через которые приходится проходить человеку, выполняющему работу. Он должен знать: что он хочет получить на выходе; как минимум одну последовательность действий, приводящую к успешному результату; где ему найти все объекты, участвующие в процедуре; как определять годность объектов к использованию; как управляться с объектами.

Указанные выше методики предназначены для ускорения выполнения этих этапов. Рассмотрим их более подробно.

При *непосредственном манипулировании* пользователь не отдает команды системе, а манипулирует объектами. В этом случае исчезает временной отрезок, в течение которого отдается команда. Для этого нужен исходный объект, которым можно манипулировать (ИОБ), и управляющий объект (УОБ), который позволяет что-то делать с ИОБ. Например, ИОБ – это иконка файла на экране, УОБ – это корзина. Тогда с помощью указателя мыши (УКМ) можно поместить ИОБ в УОБ, что эквивалентно удалению файла, но при этом всегда остается возможность восстановления файла. Другим хорошим примером является перетаскивание с помощью УКМ файла из одной папки в другую, не раскрывая целевой папки, как при выполнении операции копирования.

Снижение воздействия *потери фокуса* работников способно повысить эффективность этой деятельности. При этом специфика ситуации заключается в том, что от самих прерываний, как правило, избавиться либо трудно, либо невозможно. В таких условиях снизить их влияние можно, лишь облегчив возвращение работников к прерванному действию.

Для продолжения работы пользователь должен знать:

- на каком шаге он остановился;
- какие команды и параметры он уже дал системе;
- что именно он должен сделать на текущем шаге;
- куда было обращено его внимание на момент отвлечения.

Система должна знать, что такое «шаг». При человеко-машинном взаимодействии шагом является инициализация какого-то визуального элемента, т. е. система должна всегда знать, какой очередной визуальный элемент нужно инициализировать. Кроме этого системе должны быть известны значения введенных команд и параметров. Чтобы знать, что должен пользователь сделать на текущем шаге, системе должны быть известны основные пути прохождения визуальных элементов пользователем при выполнении своих работ.

При *ограничении принятия решения* необходимо, чтобы программа сама во многих случаях принимала решение за пользователя. Возникает вопрос, в каких случаях программа может принять решение за пользователя. При решении конкретных задач пользователем с помощью приложения обязательно существует несколько путей прохождения визу-

альных элементов. Введем следующие обозначения: ВЭ – визуальный элемент, РВЭ – раскрытие визуального элемента, ВБВЭ – выбор визуального элемента, ПРЗ – путь решения задачи, РБВЭ – работа с визуальным элементом. Некоторый n -ый путь решения задачи (ПРЗ $_n$) можно обозначить следующим образом: ПРЗ $_n = ВБВЭ_{n1}, [РВЭ_{n2}], РБВЭ_{n3}, \dots, [РБВЭ_{nm}]$. Элемент, заключенный в квадратные скобки, может отсутствовать, например, если это окно редактирования, потому что его не надо раскрывать. Такой элемент, как ComBoBox, надо раскрывать для последующего выбора нужного визуального элемента. Чтобы принимались решения за пользователя, должны быть известны все пути, соответствия путей определенным задачам, а также необходимый набор вводимых параметров и команд.

Закон Хика говорит о том, что чем меньше групп альтернативных вариантов, тем быстрее пользователь может выбрать нужную альтернативу. В этом случае программа пользовательского интерфейса должна подсказывать, предлагать пользователю, какие использовать визуальные элементы.

Разработка модели самообучения программы при человеко-машинном взаимодействии.

Направления самообучения систем: различение типов пользователей (РТП); адаптация к типам пользователей (АТП); адаптация к проблемной области (АПО); адаптация к основным способам действий пользователей (АДП).

Рассмотрим понятия «модели» и «моделирования» применительно к разработке модели самообучения.

Модель есть абстрактное представление реальности в какой-либо форме, предназначенное для отображения определенных аспектов этой реальности и позволяющее получить ответы на изучаемые вопросы [1].

Модель – это материальный или воображаемый объект, который в процессе познания замещает реальный объект, сохраняя при этом его существенные свойства. Моделирование – это процесс исследования реального объекта с помощью модели. Исходный объект называется при этом прототипом или оригиналом [2].

Общий процесс моделирования можно представить следующим образом.

1. Для выбранного объекта моделирования и задачи:

1.1. Определить цели моделирования.

1.2. Выделить свойства объекта, которые нужно моделировать.

1.3. Выбрать средства моделирования.

1.4. Построить модель.

2. Для построенной модели и поставленной задачи:

2.1. Реализовать проведение исследований на модели.

2.2. Провести анализ результатов исследований.

2.3. Если в результате процесса моделирования для поставленной задачи исследование не дало результатов, то в случае возможности модифицировать модель и перейти к пункту 2.

Цели моделирования вытекают из формулировки задачи: увеличение скорости работы пользователя; хорошее понимание пользователем последовательности шагов при работе.

Выделим свойства объекта, которые нужно моделировать. Свойства объекта «Процесс самообучения системы при работе с ней»: НП, КПФ, ОПР, ЗХ, РТП, АТП, АПО, АДП.

Средством моделирования является алгоритмический язык С#.

Для реализации целей нужно создать следующие компоненты: программу самообучения; программу слежения за действиями пользователя; модули реализации дополнительных действий.

Поток данных в модели самообучения представлен на рис.1, на котором показан: интерфейс. С интерфейсом взаимодействуют пользователь, модули реализации дополнительных действий и программа слежения за действиями пользователя. Программа самообучения в результате взаимодействия с программой слежения за действиями пользователя обучается различным дополнительным действиям, которые выполняются модулями реализации дополнительных действий.

Рассмотрим приведенные на рис. 1 элементы, которые влияют на самообучение интерфейса приложения при диалоговом взаимодействии.

Программа слежения за действиями пользователя предназначена для сравнения последовательности выбора визуальных элементов со стандартной последовательностью, предусмотренной в типичных ситуациях выполнения некоторой задачи.

2. Результаты и их обсуждение

Для примера рассмотрим типичную последовательность действий пользователя при создании нового документа в приложении Word. Покажем эту последовательность в следующем виде <Имя типового процесса>: последовательность обращений к визуальным элементам.

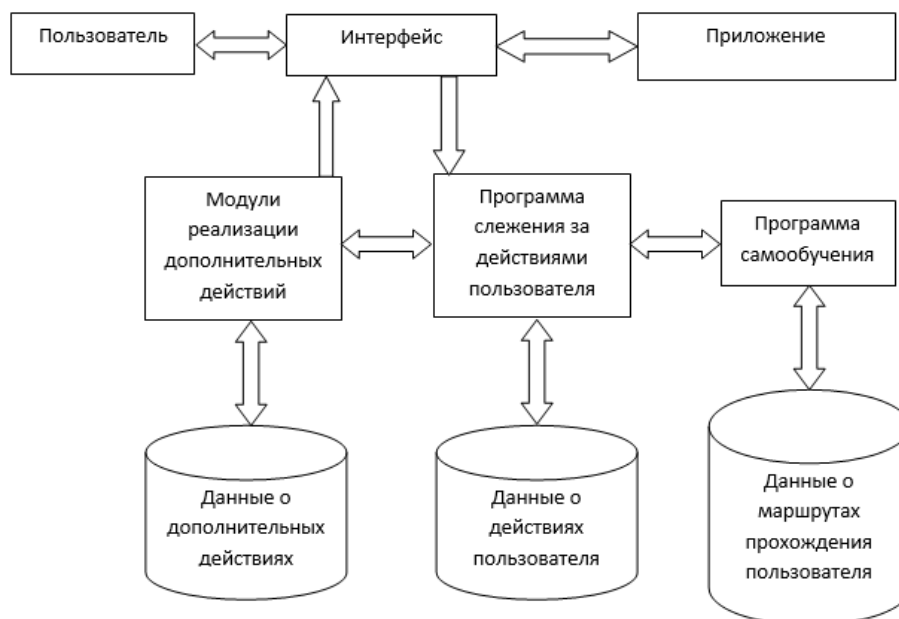


Рис. 1. Поток данных в модели самообучения

В последовательности отображается действие (Вб – выбрать, Нж – нажать, Вб&Нж – выбрать и нажать, Сзд – создать, Рб – работа с документом), название визуального элемента (Вэ), вид визуального элемента (К – кнопка, ЭМ – элемент меню, Рст – рабочий стол, ЭСmb – элемент ComboBox). Такое представление характеризует теоретический аспект последовательности выбора того или иного визуального элемента.

Пример самообучения программы при человеко-машинном взаимодействии

<Создать документ в приложении Word, содержащий текст>:

Вб(Кнопка "Office", К) → Нж(Кнопка "Office", К) → Вб(Создать, ЭМ) → Нж(Создать, ЭМ) → Вб&Нж(Microsoft Office Online, ЭМ) → Вб&Нж(Новый документ, ЭМ) → Вб(Создать, К) → Нж(Создать, К) → Сзд(Документ, Рст) → Рб(Документ, Рст) → Вб(Кнопка "Office", К) → Нж(Кнопка "Office", К) → Вб(Сохранить как, ЭМ) → Нж(Сохранить как, ЭМ) → Вб(Имя

файла, ЭСmb) → Нж(Имя файла, ЭСmb) → Вб(Тип файла, ЭСmb) → Нж(Тип файла, ЭСmb) → Вб(Имя папки, ЭСmb) → Нж(Сохранить, К).

Для хранения такой информации в базе данных предлагается табличный вариант в следующей форме.

При таком представлении несложно сопоставлять действия пользователя при выборе визуальных элементов при условии, что программа слежения за действиями пользователя сможет идентифицировать нужный процесс.

В этом случае появляется возможность модифицировать и создать новый процесс на базе стандартного, учитывающего особенности пользователя и поместить его в данные о дополнительных действиях пользователя.

С помощью модулей реализации дополнительных действий можно некоторые действия пользователя выполнять за него или предусматривать возможные следующие его действия.

Таблица 1

Данные о действиях пользователя

Имя процесса	Имя визуального элемента	Тип визуального элемента	Действия, связанные с визуальным элементом	Имя подчиненного визуального элемента	Тип подчиненного визуального элемента	Действия, связанные с подчиненным визуальным элементом
Создать документ в приложении Word, содержащий текст	Кнопка "Office"	Кнопка	Нажатие на кнопку	Меню, соответствующее нажатию кнопки «Кнопка "Office"»	Меню	Выбор элемента меню
	Создать	Элемент меню	Нажатие на элемент меню			

и т. д.

В БД хранится совокупность правил необходимых для решения указанной задачи, объединяя и выполняя их в нужной последовательности.

Правила имеют фактические данные о действиях пользователя, которые выбираются из БД. Программа слежения за действиями пользователя, накапливает новые факты о действиях пользователя.

Структура таблицы данных о дополнительных действиях аналогична табл.1.

Программа самообучения предназначена для формирования профилей пользователей, что для приложений означает выявление уровня их подготовки и выбора способа реакции на их действия. В этом случае программа самообучения, используя правила должна уметь составлять маршруты прохождения пользователя, исключая ряд пунктов для ускорения и упрощения работы пользователя.

В базе данных для программы самообучения должны содержаться следующие данные структура интерфейса.

Маршруты прохождения связаны с данными о действиях пользователя, но должны еще храниться сведения о правильности выбора визуального элемента и мерах о предотвращении неправильного выбора.

На верхнем уровне структуру интерфейса можно представить в следующем виде $Имя = \{VЭ_1 = \{VЭ_{11}, VЭ_{12}, \dots\}, VЭ_2 = \{VЭ_{21}, VЭ_{22}, \dots\}, \dots, VЭ_i = \{VЭ_{i1}, VЭ_{i2}, \dots\}, VЭ_{ij} = \{VЭ_{ij1}, VЭ_{ij2}, \dots, VЭ_{ijk} = \{\dots, \dots\}, \dots\}, VЭ_N = \{VЭ_{N1}, VЭ_{N2}, \dots\}\}$. Имя - это имя приложения или задачи, реализуемой с помощью этого приложения.

Например, для приложения Word структуру интерфейса можно представить так: Word = {(Кнопка Office, К) = {(Создать, ЭМ), (Открыть, ЭМ), (Сохранить, ЭМ), ...}, (Панель быстрого доступа, П) = {(Сохранить, К), (Отменить, К), (Повторить ввод, К), ...}, Главная = {Вставить, Вырезать, Копировать, Формат по образцу, Шрифт, Размер шрифта, ...}, Вставка = {Титульная страница, Пустая страница, Разрыв страницы, Таблица = {Вставка таблицы, Вставить таблицу, Нарисовать таблицу, ...}, Рисунок, ...}, Разметка страницы = {Темы, Цвета темы, Шрифты темы, Эффекты темы, Поля, ...}, Ссылки = {Оглавление, Добавить текст, Обновить таблицу, Вставить сноску, ...}, Рассылки = {Конверты, Наклейки, Начать слияние, ...}, Рецензирование = {Правописание, Справочники, Тезаурус, Перевод, ...}, Вид = {Разметка страницы, Режим чтения, Веб-документ, ...}, Надстройки = {Requirements, ...}, Справка = {Новые возможности, Получение справки, ...}}.

В табличном виде в базе данных структуру интерфейса можно представить, например, как в табл. 2.

В БД хранятся следующие виды правил: обобщенные правила, метаправила, формализованные правила.

Предполагается, что обобщенные правила созданы. Для получения формализованных правил из обобщенных правил используются метаправила более низкого уровня.

Таблица 2

Структура интерфейса приложения Word

Имя визуального элемента	Тип визуального элемента	Имена подчиненных визуальных элементов	Тип подчиненного визуального элемента	Действия, связанные с подчиненным визуальным элементом
Главное окно приложения Word	Окно	Главная	Панель	Выбор элемента панели
		Вставка	Панель	Выбор элемента панели

Рассмотрим примеры правил, используя вышеизложенный пример для определенного профиля пользователя. Примером обобщенного правила является такое правило «Если создать новый документ и работать с ним, то необходимо назвать имя файла; сохранить его в определенной папке».

Примером метаправила является следующее правило «Если в правиле имеется обобщенный термин «создать новый документ», то использовать конкретное имя определенного файла и сохранить его в папке с определенным именем».

В соответствии с обобщенным правилом и метаправилом получим такое формализованное правило «Вб&Нж(Кнопка «Office») → Вб&Нж(Создать, ЭМ) → Рб(Документ, Рст) → Вб&Нж(Сохранить как, ЭМ) → Вб&Нж(Имя файла, ЭСmb) → Вб(Имя папки, ЭСmb)».

Обобщенный алгоритм работы правил:

АРПр_1) Для определенной работы, находится соответствующая последовательность правил.

АРПр_2) Если такой последовательности не найдено, то начинается классическая работа пользователя с программным обеспечением и автоматическая работа модулей «Программа слежения за действиями пользователя» и т. д.

АРПр_3) Если такая последовательность найдена (профиль пользователя), то начать

процесс подбора обобщенных правил в соответствии с этой последовательностью.

АРПр_4) Если для каждого этапа последовательности работ не найдены совокупности обобщенных правил, то задействуя БД и другие модули завершить формирование совокупности обобщенных правил.

АРПр_5) Для каждого обобщенного правила сформировать совокупность метаправил, используя данные.

АРПр_6) Для определенных метаправил сформировать совокупность формализованных правил.

АРПр_7) Для формирования документа выполнить в нужной последовательности формализованные правила.

АРПр_8) Завершить работу.

Заключение

В данной статье рассмотрены подходы, позволяющие с помощью программы самообучения для автоматизированной системы реализовать одну цель – увеличение скорости работы пользователей.

Рассмотренные подходы для данной цели можно перенести на другие приложения, используя другие базы данных и хранилища.

Список литературы

1. Уёмов, А.И. Логические основы метода моделирования / А.И. Уёмов. – Москва: Мысль, 1971. – 311 с.
2. Тольяттинский государственный университет: лекции. – Режим доступа: http://edu.tltsu.ru/sites/sites_content/site216/html/media96435/lec1-2.pdf

© Д.В. Блувштейн

Ссылка для цитирования:

Д.В. Блувштейн. К вопросу о создании моделей самообучения программ при человеко-машинном взаимодействии // Инженерно-строительный вестник Прикаспия : научно-технический журнал / Астраханский государственный архитектурно-строительный университет. Астрахань : ГАОУ АО ВО «АГАСУ», 2020. № 4 (34). С. 121–126.

УДК 004.9: 69.003

DOI: 10.35108/isvp20204(34)126-133

МЕТОДИКА ОПРЕДЕЛЕНИЯ КОЭФФИЦИЕНТА СТЕСНЕННЫХ УСЛОВИЙ ТРУДА НА ОБЪЕКТАХ КАПИТАЛЬНОГО СТРОИТЕЛЬСТВА

В.С. Гладких

Пермский национальный исследовательский политехнический университет, г. Пермь, Россия

Целью работы была разработка методики определения коэффициента стесненных условий труда, позволяющей учитывать сочетания различных факторов сложности. В качестве методов использованы иерархические механизмы комплексного оценивания, представляющие собой дерево критериев, описывающее последовательность обобщения частных критериев, в узлах которого расположены матрицы свертки, отражающие правила агрегирования частных критериев. В качестве результата получена оригинальная формула, позволяющая осуществлять комплексное оценивание таких разнородных критериев как масса, количество и габариты инвентаря, необходимого для подготовки рабочего места, а также трудоемкость самого процесса подготовки идеального рабочего места. Показан наглядный пример применения предлагаемой методики в случае слабого освещения рабочего места. Данная методика может быть использована для управления строительными проектами и корректировки стоимости работ при наличии на объекте строительства таковых факторов.

Ключевые слова: *строительство, стесненные условия труда, факторы сложности, агрегирование, иерархические механизмы комплексного оценивания.*

METHODS OF DEFINING THE CONSTRAINED LABOUR CONDITIONS COEFFICIENT FOR MAJOR CONSTRUCTION PROJECTS

V. Gladkikh

Perm National Research Polytechnic University, Perm, Russia

The object of this study was to develop methods of defining the constrained labour conditions coefficient, with these methods taking into account interaction of various factors of difficulty. Hierarchical mechanisms of complex estimations were used as these methods, which can be represented as a tree of criteria reflecting consecutive generalisation of separate criteria. At the nodes of the tree there are indicator fold matrices reflecting separate criteria aggregation rules. As a result, an original formula was arrived at which facilitates complex estimation of such diverse criteria as mass, quantity and bulk of equipment necessary for preparing an ideally suitable workplace as well as the effort of the preparation process itself. There is an example given to illustrate implementation of the proposed methods in conditions of a weakly illuminated workplace. Should there be factors of this kind at a construction site, the methods may be used in managing construction projects and adjusting operation costs.

Key words: *construction, cramped working conditions, inhibiting factors, aggregation, hierarchical rating and controlling mechanisms.*

Введение

Строительно-монтажные работы могут выполняться при различных условиях, в том числе нестандартных, при которых работы осуществляются в замедленном темпе, или возрастает сложность выполнения работ, это может быть по причине необходимости под-

готовки рабочего места, например, из-за слабого освещения, загроможденности помещения оборудованием или мебелью и т.п. Такие условия принято называть стесненными, а факторы, приводящие к задержкам работы – факторами, стесняющими условия труда.

Например, по данным Макашова Н.В. в объектах реконструкции при производстве работ