



ИНТЕЛЛЕКТУАЛЬНЫЕ ПРОИЗВОДСТВА КАК РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Показано, что интеллектуальные информационные системы могут иметь два варианта: автоматизация работающих производств (концепция – системный анализ); построение новой системы (концепция – киберфизическая система). В первом варианте основной процедурой является декомпозиция системы, во втором – интеграция элементов в работающую систему. Технологии создания систем в значительной степени совпадают. Существенные отличия имеются на этапе построения структуры системы. В любом варианте важнейшим вопросом является обеспечение обмена данными между элементами системы. В работе описаны имеющиеся методы обмена, разделенные на две группы: для одноранговой технологии и технологии клиент-сервер. Проведен детальный анализ методов обмена для систем, использующих технологию клиент-сервер. Описана программная реализация интеллектуальной производственной системы управления.

Интеллектуальная автоматизированная система, производство, киберфизическая, методы обмена информацией, прикладное описание, реализация.

Настоящая работа является продолжением и развитием публикации [1]. Интерес к интеллектуальным производственным системам появился еще в 90-е года прошлого века [2].

Следует отметить, что такие системы могут рассматриваться в двух вариантах (рис. 1):

- 1) автоматизация существующих систем с ручным управлением;
- 2) построение новых производственных систем.

До недавнего времени рассматривался преимущественно первый вариант [3, 4]. Строились они на концепции системного анализа. Его задача заключалась в структуризации (декомпозиции) системы, построении технологии математического описания и программной реализации (рис. 1). Автором была предложена методология структурно-алгоритмического моделирования [2]. Она позволила сформировать однородный метод математического описания системы, базирующийся на динамическом линейном программировании [1, 4].



Рис. 1. Технология построения производственных систем

В настоящее время проявляется все больший интерес ко второму варианту [5–7]. В ее основе лежит новая концепция, получившая название «киберфизическая система» (КФС).

Киберфизическая система (англ. cyber-physical system CPS) – информационно-технологическая концепция, подразумевающая интеграцию вычислительных ресурсов в физические сущности любого вида, включая биологические и рукотворные объекты [5]. КФС охватывает как технологические (КФТС), так и производственные (КФПС) процессы и потому находит все более широкое применение для построения цифровой экономики.

Первоначально КФС развивались для технологических процессов. Задача ставилась так: имеется некий технологический процесс; заданы элементы системы (датчики, базы данных, вычислительные элементы), которые необходимо связать (интегрировать) в функционирующую систему управления.

Выделяется частный случай таких систем – интернет вещей. «Вещи» не снабжены электронным выходом и его надо сформировать перед построением системы.

Одновременно в КФТС возникает проблема хранения и работы с большими объемами данных, обрабатываемых с высокой скоростью.

Позднее КФС распространились на производственные системы, когда необходимо построить систему с нуля [2, 6, 7].

Технология формирования таких систем представлена на рисунке 1. Она похожа на технологию процедуры автоматизации, однако имеет и отличия. Структура системы не выявляется, а формируется первоначально путем деления технологического процесса на части. Каждой из них сопоставляется элемент нижнего уровня управляющей части. Затем определяются элементы более высоких уровней управляющей части. Для формального описания элементов полученной структуры может использоваться однородный метод.

Этапы реализации вариантов систем также не имеют серьезных отличий.

Следует отметить, что системы обоих вариантов являются распределенными.

Постановка задачи. В то же время применение киберфизических систем сопряжено с сетевыми про-

блемами. В последнее время появилось значительное количество разновидностей реализации распределенных систем. В связи с этим следует провести анализ разновидностей, определить их возможности и сферы применения.

Решение задачи. Распределенная система – совокупность логически интегрированных и территориально распределенных элементов, языковых, программных и технических и организационных средств, предназначенных для формирования и хранения данных, обработки информации, выработки и реализации решений [8–14].

Теоретические положения

По объекту реализации классификация систем имеет вид, показанный на рисунке 2, а по прикладному объекту – на рисунке 3.



Рис. 2. Классификация по объекту реализации

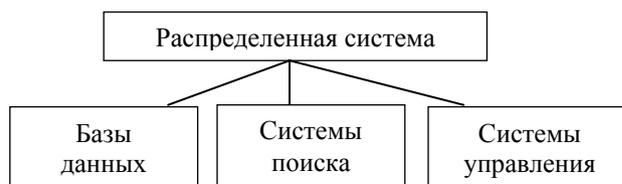


Рис. 3. Классификация по прикладному объекту

Тогда понятие «распределенная система» может быть представлено в виде схемы, показанной на рисунке 4.

Распределенная система обработки информации – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации. Ее теоретической базой служат распределенные вычисления – раздел теории вычислительных систем, изучающий теорию организации распределенных систем.

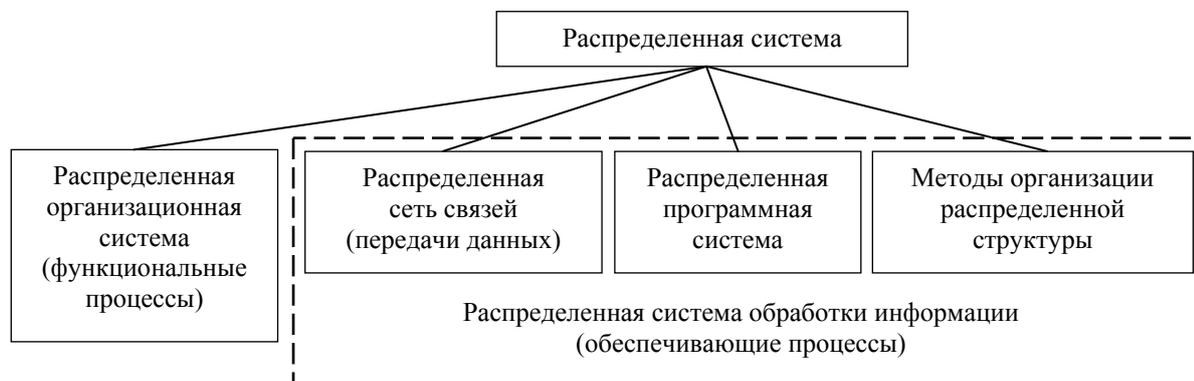


Рис. 4. Состав распределенной системы

Они являются разновидностью параллельных вычислений, предусматривающих разделение задачи на одновременно рассчитываемые подзадачи с целью максимального повышения скорости вычислений. В распределенных системах параллельные вычисления должны учитывать характеристики имеющихся вычислительных средств, возможную несинхронность вычислений и значительную временную задержку при обмене данными между узлами системы. Узлами могут быть компьютеры, процессоры, процессоры. В случае одного компьютера должна быть разделяемая память и разделение процессов (поток).

Распределенная организационная система – интерактивная автоматизированная система, используемая для различных видов деятельности при принятии решений (в том числе и многокритериальных) в ситуациях, когда невозможно или затруднительно иметь автоматическую систему из-за слабой структурированности или неструктурированности решаемых задач.

Прикладные вопросы

Существуют многочисленные способы обмена информацией, отличающиеся назначением и возможностями. Основными из них являются [14]:

- клиент-сервер;
- объектные системы;
- агентные технологии;
- компонентные системы;
- сервис-ориентированная архитектура;
- веб-сервисы;
- технологии одноранговых сетей;
- технологии Грид;
- облачные вычисления.

Сравнительные характеристики рассмотренных технологий приведены в таблице: видно, что здесь отражены технологии как с одноранговой архитектурой, так и архитектурой клиент-сервер.

Пять последних технологий достаточно сложны для локальных систем и более подходят для крупномасштабных глобальных поисковых одноранговых систем.

Таблица

Сравнительные характеристики технологий распределенной структуры

Способ	Достоинства	Недостатки
1	2	3
Web	Простота структуры	Сложность обеспечения безопасности данных. Применение в основном для поисковых систем
Модель «клиент-сервер»	Простота структуры. Возможность применения в системах управления	Трудность определения безопасности. Применение для систем относительно небольшой размерности. При росте количества клиентов повышаются требования к мощности сервера и пропускной способности канала. Устойчивая работа всех клиентов зависит от загрузки и функционирования одного сервера
Объектные системы	Простота разработки распределенных приложений в сравнении с клиент-серверным подходом. Возможность разработки приложений для гетерогенных вычислительных сред с помощью виртуальных машин и независимого описания интерфейсов взаимодействующих компонентов	Реализация удаленных вызовов гораздо сложнее реализации вызовов локальных процедур. Проблема передачи данных с адресного пространства с одной машины на другую. Неоднородность языков программирования и операционных сред: структуры данных и структуры вызова процедур. Строгая ограниченность данной технологии платформой Java. Ограниченность масштабируемости
Агентные технологии	Возможность доступа к разным точкам источника данных	Ряд серьезных проблем, с безопасностью агентных платформ. Трудности определяет принцип работы технологии: любой администратор вычислительной сети не позволит агентам путешествовать по компьютерам пользователей
Компонентные системы	Пригодность для разработки крупных систем. Простая быстрая процедура инсталляции. Увеличение доли повторного применения кода. Снижение стоимости программного обеспечения. Унификация обработки объектов различной природы. Процесс создания программного обеспечения меньше зависит от человека	Высокие требования при разработке компонент. Ограниченность технологии платформой Java

Сервис-ориентированная архитектура	Пригодность для разработки средних по размерам систем. Увеличение доли повторного применения кода. Снижение стоимости программного обеспечения. Унификация обработки сервисов различной природы	Недостаточная апробация программного обеспечения. Нет возможности работы с состояниями
Веб-сервисы	Пригодность для разработки средних по размерам систем. Увеличение доли повторного применения кода. Снижение стоимости программного обеспечения. Унификация обработки сервисов различной природы. Обеспечение безопасности данных	Достаточно сложная архитектура
Технологии одноранговых сетей	Упрощается поддержка масштабируемости при значительном росте количества узлов в вычислительной сети. Повышается отказоустойчивость сети. Возможность объединения ресурсов отдельных участников сети при возможности предоставления услуг другим участникам. Отсутствие зависимости от централизованных сервисов и ресурсов. Пригодность для систем с большим объемом данных и обменом информацией между пользователями	Рост требований к производительности каждого компьютера. Низкая степень защищенности машин из-за открытого доступа. Трудность учета гетерогенности аппаратного и программного обеспечения ее потенциальных участников. Сложность поиска доступных ресурсов без использования централизованной точки управления. Затруднения в опросах безопасности предоставления ресурсов в основном индивидуальным пользователям
Технологии Grid	Технологии Grid позволили объединить крупные комплексы обработки и хранения данных, используя следующие свойства: 1. Гетерогенность. 2. Масштабируемость. 3. Приспособляемость	Сложная инфраструктура, которую целесообразно применять для сверхбольших систем
Облачные вычисления	Сокращение в три-пять раз стоимости бизнес-приложений и более чем в пять раз стоимости приложений для конечных пользователей. Оплата по мере использования ресурса. Масштабируемость и гибкость. Высокий уровень загрузки оборудования и снижение затрат на его эксплуатацию	Сложности организации репликации данных между сервисами. Сложность администрирования в процедуре функционирования. Сложность архитектуры. Высокие риски в защите информации

Программная реализация

Иллюстрируем использование методов на примере интеллектуальной системы управления [1, 15].

Целью ее функционирования в общем случае является функционирование в традиционном (стационарном) и интеллектуальном (оперативный переход на выпуск новой продукции) режимах.

Структура распределенной организационной производственной системы представлена на рисунке 5.

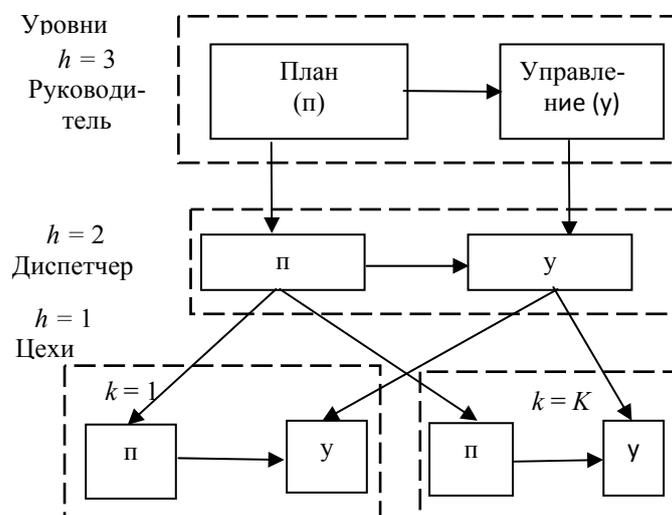


Рис. 5. Типичная структура трехуровневой системы

Детальное математическое описание иллюстрируемой системы дано в [1]. Для полноты картины в приложении приведено описание уровней диспетчера и руководителя в традиционном режиме.

К методам обмена информацией предъявим следующие требования:

1. Реализация режима клиент-сервер, что соответствует рисунку 5.
2. Согласование с базами данных (объектно-ориентированными, реляционными, NoSQL, например, SQLite) как хранилищами больших объемов информации.
3. Простота программной реализации.
4. Согласованность с языками программирования обработки данных, предпочтительно Java.

Первому требованию соответствуют пять первых способов (таблица).

Агентные технологии ограничены в применении в связи с потенциальным «шпионским» характером.

Определенный интерес представляют технологии клиент-сервер в объектно-ориентированном варианте. В то же время такая реализация часто связана с недостаточной размерностью СУБД. Объектные и компонентные системы, требуют формирования интерфейсов, что может оказаться непростой задачей. Предпочтительной представляется компонентная архитектура, более похожая на объектно-ориентированный вариант.

Схема примера приведена на рисунке 6. В качестве СУБД использовалась СУБД MySQL, сервера приложений – Apache. Работа клиентов осуществлялась на языке программирования Java.

Числовые данные получены с помощью генератора данных (язык Java), описанного в работе [1].

Приведем некоторые результаты работы модели системы.

Экранные формы для результатов решения задачи руководителя аналогичны формам на рисунке 7.

Приложение. Рассмотрим описание процесса планирования.

Уровень управления $h = 3$.

$$\begin{aligned} \mathbf{P}(T) &\geq \mathbf{R}(T), \mathbf{P}(t_i) \\ &= \mathbf{P}(t_{i-1}) + \mathbf{p}(t_i), \end{aligned} \quad (1)$$

$$\mathbf{z}(t_i) = \mathbf{A}\mathbf{z}(t_{i-1}) + \mathbf{B}\mathbf{p}_1(t_{i-1}), \mathbf{z}(0) = \mathbf{z}_0, \quad (2)$$

$$\mathbf{p}(t_i) = \mathbf{C}\mathbf{z}(t_i), \quad (3)$$

$$\mathbf{D}\mathbf{p}_1(t_i) \leq \mathbf{b}(t_{i-1}), \quad (4)$$

$$G(\mathbf{P}(T)) = \mathbf{F}\mathbf{P}(T) \rightarrow \max, \quad (5)$$

где \mathbf{z} , \mathbf{p} , \mathbf{P} – векторы незавершенного производства, планов текущего и с накоплением; \mathbf{p}_1 – вектор запуска комплекта материалов в производство; \mathbf{R} – вектор спроса; \mathbf{D} – матрица норм расходов; \mathbf{b} – вектор имеющегося количества ресурсов; \mathbf{F} – вектор прибыли от выпуска единицы продукции; \mathbf{A} , \mathbf{B} , \mathbf{C} – матрицы соответствующих размерностей; T , $t_i = iv$ – интервалы времени; $i = 1, N$; $T = Nv$.

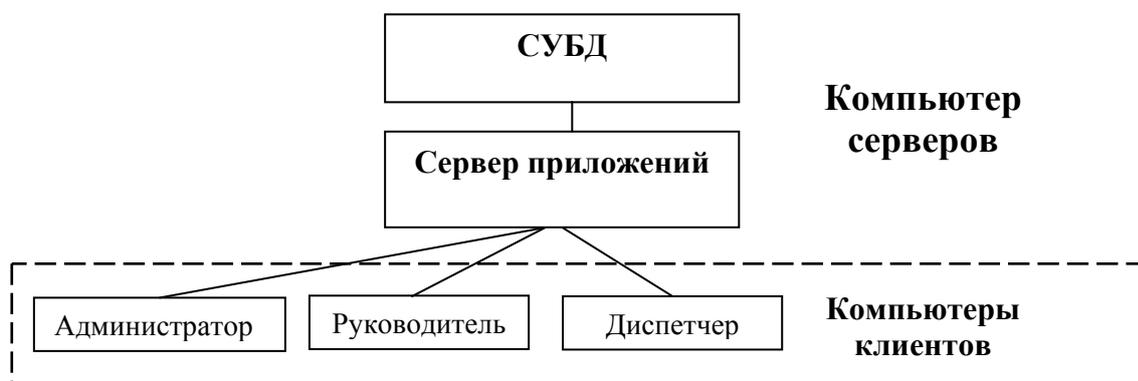


Рис. 6. Схема примера компьютерной реализации

Назад к выбору расчета									
out_3_4_date-12-11									
M	N	C, CX	DN	DV	BN	BV	Матрица A	План	
4	4	0, 2, 2, 2, 2	512, 512, 512, 512	512, 512, 512, 512	8192, 8192, 8192, 8192	8192, 8192, 8192, 8192	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	2048, 0, 0, 0	
Решить									

Рис. 7. Информация о конкретной задаче диспетчера

Уровень управления $h = 1$.

$$\mathbf{z}_k(t_i) = \mathbf{A}_k \mathbf{z}_k(t_{i-1}) + \mathbf{B}_k \mathbf{p}_{1k}(t_{i-1}), \mathbf{z}_k(0) = \mathbf{z}_{k0}, \quad (6)$$

$$\mathbf{p}_k(t_i) = \mathbf{C}_k \mathbf{z}_k(t_i), \quad (7)$$

$$\sum_{i=0}^{N-1} \mathbf{D}_1^m \mathbf{p}_{1k}(t_i) \leq \mathbf{b}^m(0), \quad (8)$$

$$\sum_{i=0}^{N-1} \mathbf{p}_k(t_i) < \mathbf{P}(T), \quad (9)$$

$$\mathbf{D}_k^\psi \mathbf{p}_{1k}(t_i) \leq \mathbf{b}_k^\psi(t_i), \quad (10)$$

$$\mathbf{D}_k^m \mathbf{p}_{1k}(t_i) \leq \mathbf{b}_k^m(t_i), \quad (11)$$

$$\mathbf{b}_k^\psi(t_i) = \mathbf{b}_k^\psi(t_{i-1}) + \Delta \mathbf{b}_k^\psi(t_{i-1}), \quad (12)$$

$$G_k(\mathbf{p}_k(t_i)) = \mathbf{F}_k \mathbf{p}_k(T) \rightarrow \max, \quad (13)$$

$$i=0, N-1, t_i = iv, t_0 = 0, T = Nv,$$

где \mathbf{z} , \mathbf{p} – вектор-столбцы (планового) незавершенного производства и ежедневного плана; \mathbf{p}_1 – вектор-столбец запуска комплектов ресурсов в производство; \mathbf{R} – вектор-столбец спроса; \mathbf{D} – матрица норм расходов ресурсов; \mathbf{b} – вектор-столбец наличного количества ресурсов; $\mathbf{b}^m(0)$ – вектор количества материальных ресурсов, которыми располагает уровень $h=3$; $\Delta \mathbf{b}$ – вектор поступления ресурсов; \mathbf{P} – вектор-столбец плана уровня $h=3$; \mathbf{F} – вектор-строка прибыли от выпуска единицы продукции; \mathbf{A} , \mathbf{B} , \mathbf{C} – матрицы, отражающие динамику процесса планирования; t_i , T – минимальный интервал времени и время моделирования; $m = 1$, M – виды материальных ресурсов; $\psi = 1$, Ψ – виды прочих ресурсов; $i = 1$, N – моменты времени; $k = 1$, k – номер подразделения.

Уровень управления $h = 2$.

Выражения (11), (13) в описании уровня $h=1$ заменяются на выражения (14), (15)

$$\mathbf{D}_k^m \mathbf{p}_{1k}(t_i) \leq \mathbf{p}_{1,k-1}^m(t_{i-1}), \quad (14)$$

$$\sum_{i=1}^N G_k(\mathbf{p}_k(t_i)) \rightarrow \max. \quad (15)$$

Одним из эффективных направлений решения проблемы цифровой экономики является построение интеллектуальных автоматизированных производственных систем управления.

Если ранее акцент делался на автоматизации существующих систем управления, работающих в ручном режиме, то сейчас все большее внимание обращают на создание новых систем.

В последнем случае в качестве концепции используют киберфизические системы (КФС). КФС предполагает не декомпозицию системы, как в случае автоматизации, а интеграцию отдельных элементов в функционирующую систему.

Технологии построения обеих разновидностей систем в значительной степени похожи. Отличия имеются в рамках определения структуры (соответственно идентификация и проектирование). В обеих разновидностях таких распределенных систем важнейшим моментом является обмен информацией. Рассмотрены методы обмена как для одноранговых систем, так и систем, использующих технологию клиент-

сервер. Сформулированы требования к методам обмена и проведен их анализ. На его основе построен прикладной пример.

Отметим, что полученные автором результаты хорошо укладываются в качестве начального инструментария при решении проблемы цифровой экономики [11] в рамках «INDUSTRIE 4.0».

Литература

1. Чертовской В. Д. Формирование описания адаптивной автоматизированной системы управления производством // Вестник Вологодского государственного университета. Серия: физико-математические, технические науки, науки о земле 2018, № 1. – С. 57–63.
2. Введение в теорию интегрированных САПР гибких технологий производств / Ю. М. Соломенцев, В. Я. Польшкалин, В. Д. Чертовской. Под общ ред Ю. М. Соломенцева. Москва : Машиностроение, 1991. – 592 с.
3. Чертовской В. Д. Основы теории адаптивных автоматизированных систем управления производством // Труды Всероссийского совещания по проблемам управления (ВСПУ-2019, Совещание). Москва : ИПУ, 2019, 17–20 июня. – С. 2676–2679.
4. Чертовской В. Д. Моделирование процессов адаптивного автоматизированного управления производством. Санкт-Петербург : Лань, 2019. – 216 с.
5. Фрадков А. Л. Кибернетическая физика: принципы и примеры. Санкт-Петербург : Наука, 2003. – 208 с.
6. INDUSTRIE 4.0 - умное производство будущего (Государственная Hi Tech Стратегия 2020, Германия) URL: http://json.tv/tech_trend_find/industrie-40-umnoproizvodstvo-buduschego-gosudarstvennaya-hi-tech-strategiya-2020-germaniya-20160227025801 (дата обращения: 26.11.2019. – Текст : электронный.
7. Ястреб Н. А. Индустрия 4.0: киберфизические системы, разумное окружение, Интернет вещей // Сборник научных статей. Вологда: ВоГУ, 2015. – Выпуск 2. – С. 136–143.
8. Таненбаум Э., ван Стеен М. Распределенные системы. Санкт-Петербург : Питер, 2003. – 877 с.
9. Шичкина Ю. А. Методы построения схемы и выполнения запросов в базах данных. Санкт-Петербург : Изд-во СПбГЭТУ «ЛЭТИ». 2016. – 206 с.
10. Радченко Г. И. Распределенные вычислительные системы / Г. И. Радченко. – Челябинск : Фотодожник, 2012. – 184 с.
11. Лычев А. В. Распределенные автоматизированные системы. – Петродворец: изд-во ВМИРЭ, 2007. – 249 с.
12. Чертовской В. Д. Архитектура ЭВМ и систем: основы теории. Санкт-Петербург, СПГУВК, 2011. – 307 с.
13. Косяков М. С. Введение в распределенные вычисления. Санкт-Петербург : НИУ ИТМО, 2014.
14. Карпов Л. Е. Архитектура распределенных систем программного обеспечения. Москва : МАКС Пресс, 2007. – 130 с.
15. Чертовской В. Д. Моделирование процессов адаптивного автоматизированного управления производством. Санкт-Петербург : Лань, 2019. – 216 с.

V.D. Chertovskoy

**INTELLIGENT MANUFACTURING AS
DISTRIBUTED CONTROL SYSTEMS**

It is shown that intelligent information systems can have two options: automation of operating industries (the concept - system analysis); building a new system (the concept - cyberphysical system). In the first version the main procedure is system decomposition, in the second - integration of elements into the operating system. Systems technologies are mostly the same. Significant differences appear during the phase of system structure formation. In any case, the most important issue is to ensure the exchange of data between the elements of the system. The paper describes the available methods of exchange divided into two groups: for peer-to-peer technology and client-server technology. Detailed analysis of the exchange methods for systems using client-server technology has been carried out. Program implementation of manufacturing control system is described.

Intelligent automated system, manufacturing, cyberphysical, methods of information exchange, applied description, implementation.