

УДК 621.391.8

Коновалов О. Ю., к.т.н. (Київський коледж зв'язку)

## АГЕНТНІ ТЕХНОЛОГІЇ У РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

**Коновалов О. Ю. Агентні технології у розподілених обчислювальних системах.** У роботі розглянуті питання використання агентних технологій в розподілених обчислювальних системах. Представлені підходи до розробки мультиагентних систем, принципи реалізації архітектури взаємодії агентів, методологія проектування і реалізації агентних рішень, а також спеціальні засоби для розробки агентно-орієнтованих систем.

**Ключові слова:** АГЕНТНА РОЗПОДІЛЕНА ОБЧИСЛЮВАЛЬНА СИСТЕМА, МУЛЬТИАГЕНТНА СИСТЕМА, ПРОГРАМНИЙ АГЕНТ, АВТОНОМНИЙ АГЕНТ, АРХІТЕКТУРА ВЗАЄМОДІЇ АГЕНТІВ.

**Коновалов А. Ю. Агентные технологии в распределенных вычислительных системах.** В работе рассмотрены вопросы использования агентных технологий в распределенных вычислительных системах. Представлены подходы к разработке мультиагентных систем, принципы реализации архитектуры взаимодействия агентов, методология проектирования и реализации агентных решений, а также специальные средства для разработки агентно-ориентированных систем.

**Ключевые слова:** АГЕНТНАЯ РАСПРЕДЕЛЕННАЯ ВИЧИСЛИТЕЛЬНАЯ СИСТЕМА, МУЛЬТИАГЕНТНАЯ СИСТЕМА, ПРОГРАММНЫЙ АГЕНТ, АВТОНОМНИЙ АГЕНТ, АРХИТЕКТУРА ВЗАИМОДЕЙСТВИЯ АГЕНТОВ.

**Konovалov O. Iu. Agent technologies in distributed computing systems.** This paper considers the use of agent-based technologies in distributed computing systems. Approaches to the development of multi-agent systems, the principles of the architecture of interaction between agents, the methodology of the design and implementation of agent-based solutions, as well as special funds for the development of agent-oriented systems.

**Keywords:** AGENT-BASED DISTRIBUTED COMPUTING SYSTEM, MULTI-AGENT SYSTEMS, SOFTWARE AGENTS, AUTONOMOUS AGENTS, INTERACTING AGENTS ARCHITECTURE.

**Вступ.** Розвиток розподілених обчислювальних систем дозволяє здійснювати делегування складних завдань програмним системам (агентам), що у свою чергу, дозволяє представляти і вирішувати завдання, що важко формалізуються. При проектуванні розподілених систем доступу багатоагентна технологія дозволяє поєднати в єдиній системі як протокольне, так і будь-яке прикладне програмне середовище обробки і взаємодії з різними типами даних. Така система має гнучкість, масштабованість і ефективність розподілу навантаження між серверами.

Агентні розподілені обчислювальні системи є актуальним і перспективним напрямом на сучасному етапі розвитку розподілених обчислювальних систем.

У основі агентних технологій лежать розробки засновані на таких поняттях системної архітектури як: програмний агент і автономний агент. Згідно з цими визначеннями програмний агент – це автономний процес, здатний реагувати на середовище виконання і викликати зміни в середовищі виконання, можливо, в кооперації з користувачами або іншими агентами [1], а *автономний агент* – це система, що знаходиться усередині оточення і є його частиною, сприймає це оточення (його сигнали) і що впливає на оточення для виконання власної програми дій [2].

Агентна технологія містить в собі переваги декількох різних дисциплін. Вона включає достоїнства об'єктноорієнтованих і розподілених програмних середовищ, компонентної концепції розробки програмного забезпечення і навіть штучного інтелекту і експертних систем. Агентні технології можуть стати успішним архітектурним рішенням саме тому, що вони здатні інтегрувати ці різні аспекти і виявляти їх внутрішній потенціал.

Набагато ширше агентні технології застосовуються у сфері комунікації, особливо у сфері мобільних пристроїв. Крім того, агентна концепція зручна для створення розподілених систем [3], у тому числі і доступу до інформаційних ресурсів.

Написання агентно-орієнтованої програми виливається в створення набору агентів, які разом забезпечують усю необхідну функціональність. Мобільні агенти відповідають за активні частини програми, тоді як системні агенти забезпечують доступ до локальних служб. І ті, і інші спілкуються через вбудовані механізми комунікації. Оскільки агентно-орієнтовані програми ґрунтуються на співпраці різних незалежних об'єктів, то вони структурно не

монолітні. Тому при створенні агентно-орієнтованих програм нормальною практикою стає повторне використання існуючих агентів.

При організації систем доступу кінцевого користувача до інформаційних ресурсів [3], особливо систем, орієнтованих на використання в онлайн-режимі в мережі Internet, однією з ключових технічних характеристик системи, що визначає зручність її використання, являється час відгуку на запит користувача. Саме ця характеристика і є найбільш вузьким місцем усіх розподілених систем.

Розвиток інформаційних і телекомунікаційних систем на сучасному етапі вимагає розробки теоретичних основ проектування інтегрованих систем нового покоління, що включають телекомунікаційні системи і комп'ютерні мережі або GRID-системи, в яких акумулюються розподілені інформаційні і обчислювальні ресурси. Такі системи надають своїм користувачам, як зовнішнім агентам, високоякісні послуги для їх масового віддаленого доступу і ефективного використання розподілених інформаційних і обчислювальних ресурсів за допомогою IP-протоколів і інших засобів керованого зв'язку і передачі інформації.

Вдосконалення ІТКС пов'язане передусім з розвитком методології автоматизації, адаптації і інтелектуалізації систем мережевого управління інформаційними потоками на базі динамічних моделей ТКС як складних об'єктів управління із змінною структурою, методів оптимізації процесів маршрутизації інформаційних потоків і принципів адаптивного і інтелектуального управління трафіком з використанням мульти-агентних технологій і протоколів нового покоління (IPv6 та ін.). На цьому шляху можливий як облік реальної динаміки ТКС, тобто фактичного стану або зміни структури (топології) і параметрів (вагів каналів зв'язку) ТКС в реальному часі, так і адаптація до різних чинників невизначеності на основі моніторингу і функціональної діагностики системи [4...6].

*Агенти* можуть застосовуватися при вирішенні наступних завдань: *мобільні* обчислення; *пошук* інформації; *відбір* (обробка) інформації; *моніторинг* даних; *універсальний* доступ до даних.

У контексті цієї роботи під агентом розумітимемо механізм інкапсуляції і обміну розподіленими знаннями і функціями. Кожен агент – це процес, що має певну частину знань про об'єкт і можливість обмінюватися цими знаннями з іншими агентами. Залежно від типу, агент може підтримувати і інтерфейс з користувачем. Під багатоагентною системою розумітимемо багатокомпонентну систему, що складається з агентів із специфікованим інтерфейсом.

Існуючі варіанти архітектури багатоагентних систем і раціональний вибір архітектури окремого агента і багатоагентної системи в цілому [7], істотно залежать від того, яка концептуальна модель агента, прийнятий для її опису формалізм і мова специфікацій, яка математична модель кооперації агентів при спільному функціонуванні в системі, на яке застосування або клас додатків орієнтована багатоагентна система, а також від ряду інших чинників [8]. Можна з упевненістю стверджувати, що скільки існує і розробляється агентів і багатоагентних систем, стільки існує і архітектур. Проте, можна вибрати деякі характерні варіанти, що утілюють основні принципи конструювання архітектури і які розглядаються як перспективні.

Основне призначення архітектури взаємодії агентів полягає в тому, щоб забезпечити скоординовану поведінку агентів при вирішенні загальних або своїх приватних завдань але тут можна виділити два основні варіанти архітектури. У одному з них агенти не утворюють ієрархії і вирішують загальну задачу повністю в розподіленому варіанті. У іншому варіанті координація розподіленого функціонування агентів в тій чи іншій мірі підтримується спеціально виділеним агентом, який при цьому відноситься до мета-рівня по відношенню до інших агентів. Існують і, мабуть, виникатимуть складніші, ієрархічно організовані схеми взаємодії агентів, проте обмежимося коротким розглядом названих двох варіантів.

Груба класифікація архітектури агентів ґрунтується на парадигмі, що лежить в основі прийнятої архітектури. За цією ознакою розрізняють два основні класи архітектури [9]:

- архітектура, яка базується на принципах і методах штучного інтелекту, тобто систем, заснованих на знаннях ("deliberative agent architecture", "архітектура розумного агента");
- архітектура, заснована на поведінці ("reactive architecture") або "реактивна архітектура" (заснована на реакції системи на події зовнішнього світу).

Насправді до теперішнього часу серед розроблених архітектур не існує таких, про які можна було б ясно сказати, що вони є чисто поведінковими або засновані тільки на знаннях. Будь-яка з розроблених архітектур є по суті гібридною. Кожна окремо взята організація має свою специфічну структуру, характеризується особливістю діяльності, що викликає необхідність розробки методів, моделей і програмних засобів, що дозволяють вирішувати специфічні завдання, що виникають при побудові систем. Для виділення основних можливостей, яким повинна відповідати система, що розробляється, необхідно розглянути існуючі і найбільш відомі системи, що забезпечують необхідні параметри.

При агентно-орієнтованому програмуванні досягається найбільша інкапсуляція – інкапсуляція коду, змінних і ініціалізації, за рахунок чого досягається ряд переваг в порівнянні з традиційними підходами. Наприклад, зменшення кількості реалізацій поведінки, в порівнянні з традиційними підходами [10]. Також застосування програмних агентів дозволяє понизити два найважливіші параметри – вартість впровадження і обслуговування. До того ж використання мультиагентних систем дозволяє досягти більшої адаптивності і помилкостійкості в порівнянні з системами, розробленими на підставі традиційних підходів. Мультиагентну систему (МАС) у загальному вигляді можна представити у вигляді множини з трьох елементів [10]: *Агенти*, *Середовище*, *Зв'язки між Середовищем і Агентами*.

$$\text{МАС} = \langle \text{Агенти}, \text{Середовище}, \text{Зв'язки} \rangle$$

Кожен *Агент<sub>i</sub>* описується за допомогою множини з чотирьох елементів: *Стан<sub>i</sub>*, *Вхід<sub>i</sub>*, *Вихід<sub>i</sub>*, *Процес<sub>i</sub>*:

$$\text{Агент}_i = \langle \text{Стан}_i, \text{Вхід}_i, \text{Вихід}_i, \text{Процес}_i \rangle,$$

де *Стан<sub>i</sub>* – це множина змінних, що повністю визначають агента; *Вхід<sub>i</sub>* і *Вихід<sub>i</sub>* – підмножини *Стану<sub>i</sub>*, елементи яких пов'язані з середовищем.

*Процес<sub>i</sub>* – автономний метод, що виконує відповідні зміни над *Станом*. "Автономний метод" має на увазі, що цей метод викликається без якої-небудь зовнішньої участі. *Середовищем* є множина з двох елементів:

$$\text{Середовище} = \langle \text{Стан}_c, \text{Процес}_c \rangle$$

Важливою особливістю такого представлення *Середовища* є те, що *Середовище* є само по собі активне, оскільки містить свій власний *Процес<sub>c</sub>*, який може змінювати *Стан<sub>c</sub>* незалежно від агентів, що входять в це *Середовище*. *Вхід<sub>i</sub>* і *Вихід<sub>i</sub>* різних агентів пов'язані з елементами *Стану<sub>c</sub>*, але середовище не розрізняє, які з елементів *Стану<sub>c</sub>* знаходяться в залежності з ними. Відсутність *Входу* і *Виходу* у *Середовища* означає, що воно, на відміну від *Агентів*, є необмеженим. Якби в *Середовище* був доданий *Вхід* і *Вихід*, те це означало б, що середовище обмежене і є фактично високорівневим агентом. Таким чином, може моделюватися взаємодія високорівневих агентів з низькорівневими.

Ще одним важливим аспектом розробки мультиагентної системи є вибір рішень орієнтованих на підтримку однопотоківих або багатопотокових застосувань. Досить часто цілком вистачає єдиного потоку в процесі (додатку), проте, однопоточні застосування іноді не дуже швидко реагують на запити користувача, наприклад, якщо цей потік виконує складну операцію (друк великого документу). В той же час надто багато потоків в одному процесі можуть істотно скоротити його продуктивність, оскільки процесор буде вимушений постійно перемикатися між потоками [11].

Для розробки агентно-орієнтованих систем використовують спеціальні засоби, такі як кроссплатформне програмне оточення для програмування мультиагентних систем, мультиагентні симулятори, Java бібліотеки для моделювання мультиагентних систем, набори інструментів для створення систем, заснованих на агентах, спеціальні бібліотеки і компоненти для поширених мов програмування. З найбільш відомих:

- NetLogo – кроссплатформне програмне оточення для програмування мультиагентних систем;
- VisualBots – безкоштовний мультиагентний симулятор в Microsoft Excel з Visual Basic синтаксисом;
- MASON – Java бібліотека для моделювання мультиагентних систем;
- REPAST – набір інструментів для створення систем, заснованих на агентах;
- JADE – Java бібліотека для створення мультиагентних систем (JADE в wiki);
- SemanticAgent – SWRL/JAVA;
- CogniTAO – C++ платформа розробки автономних мультиагентних систем орієнтована на реальних роботів і віртуальних істот (CGF).

Найширше використовується платформа багатоагентного програмування JADE (Java Agent Development Framework), реалізована на мові Java. JADE спрощує розробку багатоагентних систем завдяки використанню інструментів, що підтримують фази налаштування і розгортання системи. Агентна платформа може встановлюватися на комп'ютерах з різними операційними системами, і конфігуруватися через віддалений GUI – інтерфейс навіть під час виконання агентів.

Для платформи JADE розроблено середовище JADEX, яке є розширенням багатоагентної платформи. Це середовище припускає гібридну реактивно-деліберативну архітектуру, в якій агент розглядається як "чорна скринька", яка приймає і відправляє повідомлення. Грунтуючись на результатах обробки повідомлень, внутрішніх і зовнішніх подій, деліберативний механізм приймає рішення про перехід до нового плану дій або продовження старого. Діючий план може відправляти повідомлення інших агентів, змінювати умови, формувати нові цілі і викликати внутрішні події. Система використовує бібліотеку планів, які обробляються як Java-класи.

Розглянемо на прикладі Gaia [12] методологію побудови мультиагентних систем, яка передбачає побудову наступних метамodelей:

- організаційної метамodelі, яка описує архітектуру багатоагентної системи і її функції. Функції визначаються на момент завдання цілей організації і порядку їх виконання;
- метамodelі оточення, яка описує суті, що впливають на сприйняття агентів;
- метамodelі цілей і завдань, яка описує зміни станів агента в часі залежно від виконання конкретних завдань, шляхи досягнення цілей і дії, що виконуються у разі, якщо мета не може бути досягнута;
- метамodelі агента, яка описує поведінку кожного агента;
- метамodelі взаємодії, яка описує поведінку двох і більше взаємодіючих агентів.

Залежно від мови опису вона може бути представлена у вигляді UML-діаграми кооперації або AUMML-діаграми.

З точки зору розробників інформаційних систем, агент – це модуль програмного забезпечення, що виконується на певній платформі, здійснює деякі дії, такі як відображення і введення інформації, і що обмінюється сполученнями з іншими агентами або людиною [13]. Таким чином, в найпростіших випадках можна робити обгортку (wrapper) для веб-сервісу у вигляді агента (чи навпаки), в складніших (при проектуванні систем) використовувати стандарти W3C. Веб-сервіси, як і мультиагентні системи, вирішують основну проблему гетерогенних IT-середовищ по забезпеченню інтеграції і взаємодії систем. Стає усе більш зрозуміло, що агентські системи, семантична мережа і веб-сервіси є додатковими, а не конкуруючими технологіями [14].

Перевага веб-служб над попередніми middleware-технологіями на основі компонентів (наприклад, CORBA і DCOM) в тому, що протоколи є веб-орієнтованими. Отриманий у вигляді метамodelей опис багатоагентної системи може оброблятися відповідними модулями. Поведінка агентів визначається їх цілями і завданнями, а також взаємодією між ними. Цілі у свою чергу можуть розбиватися на простіші цілі, які можуть досягатися певними послідовностями завдань. Сукупність завдань і взаємодій визначає глобальну поведінку системи. На основі усього опису системи відбувається генерація коду. Отриманий

опис може бути використаний для створення програмного коду багатоагентної системи під конкретну платформу.

Таким чином, усі розглянуті програмні продукти здатні допомогти розробникові систем моделювання мережевих динамічних об'єктів з розподіленими параметрами на вищому рівні програмування.

**Висновки.** У роботі були розглянуті основні принципи функціонування агентних технологій. Розробка агентних систем придбала такі масштаби, що постало питання про стандартизацію, і незабаром з'явилися стандарти MASIF і FIPA, що дають рекомендації по створенню систем мобільних агентів і систем інтелектуальних агентів відповідно.

Методи розробки ПО (наприклад, об'єктно-орієнтований аналіз і проектування) є невідповідними для розробки агентних систем. Є фундаментальні відмінності між концепціями використовуваними об'єктно-орієнтованими розробниками і агентно-орієнтованим представленням. Методології аналізу і проектування агентських систем допомагають об'єднати різні процеси, необхідні для реалізації багатоагентної системи в єдиний ланцюжок, і долають розрив між концепціями, використовуваними об'єктно-орієнтованими і агентно-орієнтованими розробниками.

На даний момент немає єдиного уявлення про те, як треба розробляти агентні системи. Методологій багато (Gaia, MaSE, AUMML, Fusion, Tropos та ін.), кожна претендує на оптимальне рішення, принаймні, приватних питань. Різні школи і течії розрізняються не лише на рівні послідовності дій, але і на рівні прийнятих позначень.

### Література

1. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван-Стеен. – СПб.: Питер, 2003. – 877 с.
2. Franklin S., Graesser C. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents // Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings, Lecture Notes in Computer Science. – 1997. – Vol. 1193. – PP. 22-35.
3. Федотов В.Б. Построение раксределенной системы доступа к информационным ресурсам на основе багатоагентной архитектур [Електронний ресурс] / В.Б. Федотов // Режим доступу: [http://www.benran.ru/Magazin/cgi-bin/Sb\\_03/pr03.exe?!10](http://www.benran.ru/Magazin/cgi-bin/Sb_03/pr03.exe?!10)
4. Krassimir Markov, Vitalii Velychko, Oleksy Voloshin (ed.) Natural and Artificial Intelligence ITHEA® Sofia, Bulgaria [Електронний ресурс]. – 2010 // Режим доступу: [http://foibg.com/ibs\\_isc/ibs-17/ibs-17-p14.pdf](http://foibg.com/ibs_isc/ibs-17/ibs-17-p14.pdf)
5. Тимофеев А.В. Методы высококачественного управления, интеллектуализации и функциональной диагностики автоматических систем. Часть I, Часть II / А.В. Тимофеев. – Мехатроника, автоматизация, управление. – 2003. – № 5. – 2004. – № 2.
6. Тимофеев А.В. Модели и методы маршрутизации потоков данных в телекоммуникационных системах с изменяющейся динамикой / А.В. Тимофеев, А.В. Сырцев – М.: Новые технологии, 2005 г. – 85 с.
7. Michal Laclavik, Marian Babik, Zoltan Balogh, Emil Gatial, Ladislav Hluchy. Semantic Knowledge Model and Architecture for Agents in Discrete Environments n: Frontiers in Artificial Intelligence and Applications, Vol 141, 2006. Proc. of ECAI 2006 Conference, G.Brewka et al.(Eds.), IOS Press, pp.727-728. ISBN 1-58603-642-4. ISSN 0922-6389. 29 August - 1 September 2006, Riva del Garda, Italy.
8. Michal Laclavik, Marian Babik, Zoltan Balogh, Ladislav Hluchy. Agent OWL: Semantic Knowledge Model and Agent Architecture In Computing and Informatics. Vol. 25, no. 5 (2006), p. 419-437. ISSN 1335-9150, Chapters 1, 4, 5
9. Городецкий В.И. Многоагентные системы (обзор) / В.И. Городецкий, М.С. Грушинский, А.В. Хабалов // Новости искусственного интеллекта. – 1998. – №2. – С. 64-116.
10. Олійник А. О. Агентні технології в системах групової відеокommунікації /А.О. Олійник, О.О. Ключкова, О.О. Олійник // Біоніка інтелекту: наук.-техн. журнал.

[Електронний ресурс]. – 2010. – № 3 (74). – С. 83–89. // Режим доступу : [http://archive.nbu.gov.ua/portal/natural/bionint/2010\\_3/Oleynik.pdf](http://archive.nbu.gov.ua/portal/natural/bionint/2010_3/Oleynik.pdf)

11. Коваленко А.Д. Агентные технологии представления и обработки знаний в локальных и распределенных экспертных системах / А.Д. Коваленко [Електронний ресурс] // Режим доступу: [http://www.iai.dn.ua/public/JournalAI\\_2005\\_4/Razdel5/08\\_Kovalenko.pdf](http://www.iai.dn.ua/public/JournalAI_2005_4/Razdel5/08_Kovalenko.pdf)

12. Michael Wooldridge, Nicholas R. Jennings, David Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. [Електронний ресурс] // Режим доступу: <http://masters.donntu.edu.ua/2006/fvti/grach/library/st7.htm>

13. Тарасов В.Б. Агенты, багатоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте / В.Б. Тарасов // Новости искусственного интеллекта. – 1998. – №3. – С.5-54.

14. Ian Dickinson. The Semantic Web and Software Agents: Partners, or Just Neighbours? AgentLink News 15, 2004, pp. 3-6.

УДК 621.39:316.776.22

**Мурай А. В.**, асп (*Учебно-научный инст-т холода, криотехнологий и экоэнергетики ОНАПТ*)

### **ОЦЕНКА КАЧЕСТВА ТЕЛЕКОММУНИКАЦИОННЫХ УСЛУГ С УЧЕТОМ СТЕПЕНИ УДОВЛЕТВОРЕНИЯ ОЖИДАНИЙ И ТРЕБОВАНИЙ ПОЛЬЗОВАТЕЛЕЙ**

**Мурай О.В. Оцінка якості телекомунікаційних послуг з урахуванням ступеня задоволення очікувань і вимог користувачів.** Розглянуто питання оцінки якості телекомунікаційних послуг з позиції користувачів послуг, а також аспекти діяльності постачальника послуг, спрямовані на задоволення очікувань і вимог користувачів. Проаналізовано існуючі методики та підходи до оцінки якості послуг з точки зору користувачів послуг. Зроблено спробу встановлення взаємозв'язку між споживчими властивостями послуг та ступенем задоволеності вимог користувачів, запропоновано критерій оцінки якості послуг – інформаційна насиченість послуги.

**Ключові слова:** ОЦІНКА ЯКОСТІ, ПОКАЗНИКИ ЯКОСТІ, ТЕЛЕКОМУНІКАЦІЙНІ ПОСЛУГИ, ІНФОРМАЦІЙНА НАСИЧЕНІСТЬ

**Мурай А.В. Оценка качества телекоммуникационных услуг с учетом степени удовлетворения ожиданий и требований пользователей.** Рассмотрены вопросы оценки качества телекоммуникационных услуг с позиции пользователей услуг, а также аспекты деятельности поставщика услуг, направленные на удовлетворение ожиданий и требований пользователей. Проанализированы существующие методы и подходы к оценке качества услуг с точки зрения пользователей услуг. Сделана попытка установления взаимосвязи между потребительными свойствами услуг и степенью удовлетворенности требований пользователей, предложен критерий оценки качества услуг – информационная насыщенность услуги.

**Ключевые слова:** ОЦЕНКА КАЧЕСТВА, ПОКАЗАТЕЛИ КАЧЕСТВА, ТЕЛЕКОММУНИКАЦИОННЫЕ УСЛУГИ, ИНФОРМАЦИОННАЯ НАСЫЩЕННОСТЬ

**Murai O.V. Assessing of telecommunication services quality from the perspective of service users.** The problems of assessing the quality of telecommunication services from the perspective of service users, as well as aspects of the service provider activity oriented towards the expectations and demands of users are considered. The existing methods and approaches to services quality estimation in terms of service users are analyzed. An attempt is made to relate the consumer properties of services and the degree of satisfaction of user requirements. The criterion of the services quality estimation, the information saturation of service, has been proposed.

**Keywords:** QUALITY ESTIMATION, QUALITY INDEXES (INDICATORS), TELECOMMUNICATION SERVICES, INFORMATIONAL SATURATION

**Постановка проблеми, анализ исследований и публикаций.** Рост значимости информации в жизни общества, а также развитие средств ее передачи, обработки и хранения приводят к усилению роли инфокоммуникационной (информационно-телекоммуникационной) системы, как области деятельности, включающей все информационные объекты (пользователи и другие источники и потребители информации) и