Received 24 March 2015
Revised 6 August 2015
7 September 2015
Accepted 13 September 2015

On the model design of integrated intelligent big data analytics systems

Kun Chen
Department of Financial Mathematics and Engineering, South University of Science and Technology, Shenzhen, China
Xin Li
Department of Information Systems, City University of Hong Kong, Kowloon, Hong Kong, China, and
Huaiqing Wang
Department of Financial Mathematics and Engineering, South University of Science and Technology, Shenzhen, China

Abstract

Purpose – Although big data analytics has reaped great business rewards, big data system design and integration still face challenges resulting from the demanding environment, including challenges involving variety, uncertainty, and complexity. These characteristics in big data systems demand flexible and agile integration architectures. Furthermore, a formal model is needed to support design and verification. The purpose of this paper is to resolve the two problems with a collective intelligence (CI) model.

Design/methodology/approach – In the conceptual CI framework as proposed by Schut (2010), a CI design should be comprised of a general model, which has formal form for verification and validation, and also a specific model, which is an implementable system architecture. After analyzing the requirements of system integration in big data environments, the authors apply the CI framework to resolve the integration problem. In the model instantiation, the authors use multi-agent paradigm as the specific model, and the hierarchical colored Petri Net (PN) as the general model.

Findings – First, multi-agent paradigm is a good implementation for reuse and integration of big data analytics modules in an agile and loosely coupled method. Second, the PN models provide effective simulation results in the system design period. It gives advice on business process design and workload balance control. Third, the CI framework provides an incrementally build and deployed method for system integration. It is especially suitable to the dynamic data analytics environment. These findings have both theoretical and managerial implications.

Originality/value – In this paper, the authors propose a CI framework, which includes both practical architectures and theoretical foundations, to solve the system integration problem in big data environment. It provides a new point of view to dynamically integrate large-scale modules in an organization. This paper also has practical suggestions for Chief Technical Officers, who want to employ big data technologies in their companies.

Keywords Collective intelligence, System integration, Big data analytics, Model design

Paper type Research paper

1. Introduction

Imagine that an e-commerce company wants to develop an integrated big data analysis system to support its business. Every day, 200 million people visit its web site and make one million orders from a range of several millions of goods. In addition, sales

This paper was supported by GRF grant (CityU 149412) from the Hong Kong Government, SUSTC Fundamental Research Grant (FRG-SUSTC1501A-20), and the Shenzhen Research Grant (No. JCYJ2010417105742712).

Industrial Management & Data Systems
Vol. 115 No. 9, 2015
pp. 1666-1682

0263-5577
DOI 10.1108/IMDS-03-2015-0086

may increase sharply on holidays. The system must address various types of data, such as customer orders, comments, and phone calls. It is also important to monitor traffic and weather conditions in real time to ensure that deliveries are completed as soon as possible. Moreover, the system should be able to immediately address various potential emergency situations once relevant information is obtained. To this end, how does one design a system in which all participants can work cooperatively? Actually, this is a typical scenario including several so-called big data analysis tasks. However, we must go beyond simply applying Hadoop, MapReduce, and other big data techniques to design an integrated system model.

The design of a big data system differs considerably from that of a traditional database-supported decision support system (DSS) (Madden, 2012). Such a system involves more entities, data, and participants; therefore, the system has special requirements in terms of data management, model design, and quality of service (QoS), especially in a multiple subsystem integration process (as shown in Table I).

Requirement 1: an integrated big data system should be working closely with up/down value stream partners to achieve common goals through new ways of organizing data to facilitate more effective decisions. Big data analytics address large volumes and distributed aggregations of various types of data (O'Leary, 2013). The data may be from audio, video, social networks, or web forums. Big data no longer relies on databases or data warehouses. No SQL methods and stream processing in memory, are incorporated into the system. Therefore, integrating different data management mechanisms is a considerable challenge.

Requirement 2: an integrated big data system should be adapting and modifying key business processes and more quickly delivering applications. Big data analytics models are not typically predefined due to the presence of dynamic environments. Such models typically require iterative solutions for testing and improvement. Moreover, business processes in big data analytics systems should be flexible (Talia, 2013). Participants in such models include software systems, mobile devices, web services,

		Database system	Big data system
Data management	Source	Internal/defined	External/various
	Format	Structured	Structured/unstructured
	Update	Update periodically	Change every second
	Size	Megabytes or gigabytes	Petabytes or exabytes
	Storage	SQL-like DB	SQL-like DB No SQL system Memory
Model design	Analytic model	Analysis models designed against stable environment	Need to iteratively test/improve models
	Participants	Software systems within organization	Various software systems, devices, web services, and, etc.
	Business process	Predefined business logic	Dynamic business process built based on distributed functional modules
Quality of services	Time	Normal response time	Involving time-stamped events
	Reliability	Complete and reliable	Incomplete and fuzzy
	Security	High	Very high

Table I.
Database supported vs big data supported systems

and humans. Building dynamic business processes that allow for cooperation among various participants is another challenge (Liu and Lv, 2015).

Requirement 3: an integrated big data system should provide the ability to respond to business requirements quickly and accurately while reusing functional and integration components. QoS is vital in big data analytics systems (Marx, 2013). Jacobs (2009) states that "It is easier to get the data in than out." Systems occasionally need to react to an event, such as a service outage or a change in a patient's medical condition, in real time. Thus, obtaining an overview of QoS properties of the system during design is the third challenge, especially for reuse and integration of big data analytics components.

To address these challenges, we propose a model design methodology using collective intelligence (CI) for big data analytics integration. CI is the shared or group intelligence that emerges from the collaboration, collective efforts, and competition of many individuals and appears in consensus decision making (Glenn, 2015). CI systems often exhibit the following features: first, they effectively adapt in uncertain and unknown environments; second, they organize themselves autonomously; and third, they exhibit "emergent" behavior (Schut, 2010). These features are well suited for the aforementioned big-data-analytics-supported DSS. Therefore, the question of how one uses CI to integrate big data analytics systems naturally arises.

A CI model design methodology is comprised of five steps: assessment; model; simulation; verification and; validation (Schut, 2010). Modeling is further comprised of generic and specific modeling. For the generic model, we use a hierarchical colored Petri Net (HCPN) to model the DSS system. A PN is a formal model for the description of distributed systems. It is a popular theoretical basis for modeling, simulating, and analyzing concurrent systems, and it provides powerful tools for validation and verification (Wang *et al.*, 2002; Ma and Tsai, 2008; Kounev, 2005). For a specific model, we instantiate a generic model based on the multi-agent paradigm. We do this because intelligent agents are well suited to address the problem of analyzing dynamic information in an adaptive manner because of their characteristics of autonomy, social ability, reactivity, proactivity, and mobility (Dalal *et al.*, 2004). The case study has shown the practical implications of our design. On theoretical front, our work extends the current research on data management and system engineering in big data area, to provide a conceptual CI model for big data integration. In this general CI model, other practices on different system architectures and theoretical models are easily incorporated in future work.

The remainder of the paper is organized as follows. Section 2 discusses relevant prior work. Section 3 describes the multi-agent system design. Section 4 proposes a PN model for the proposed system. Section 5 describes a case study. Finally, we draw the conclusions of this research.

2. Background research

2.1 Big data analytics systems and system integration

Big data refers to the rising flood of digital data from many sources, including the sensors, digitizers, scanners, software-based modeling, mobile phones, internet, videos, e-mails, and social network communications (Berkovich and Liao, 2012). Current researches have provide various data analysis and data mining methods, as well as parallel computing infrastructures to support big data analytics tasks. For example, the core of Hadoop is a programming framework, MapReduce, which crunches data from other sources and can deliver analysis or sometimes just aggregated data to be used in other analytical systems (Raden, 2012).

Given these techniques, big data analytics systems are developed for broad business intelligence applications (Chen *et al.*, 2012). How to integrate these systems in organization becomes an important question. Although sparse, the big data integration research has been emerging from two regards. One regard is to integrate various data on semantic or schema (Dong and Srivastava, 2013). For example, Knoblock and Szekely (2015) developed a semantic mapping method to integrate data in the cultural heritage domain. Another regard is to integrate big data systems in architecture level. For example, Fernandez *et al.* (2015) developed a data integration stack with processing layer and messaging layer to support nearline and offline big data integration. Demirkan and Delen (2013) use cloud computing architecture to put analytics and big data as services, and realize the functional integration. In this paper, we want to analyze the big data analytics system integration problem in a conceptual CI framework. In the framework, system architecture is implemented with multi-agent paradigm, and theoretical model is developed based on PN for system verification.

2.2 CI

Glenn (2015) defines CI as an emergent property from synergies among three elements: data/information/knowledge; software/hardware; and experts and others with insight, who continually learn from feedback to produce just-in-time knowledge to produce better decisions compared to any of these elements acting alone. Khazaei and Lu (2014) extend a framework primarily designed to support small-scale interactions, making it applicable to research on larger online collectives.

In recent studies, CI has been used mainly for goal decomposition, distributed system design, and information collection. For example, Wolpert *et al.* (2000) use CI to configure the nodal elements of a distributed dynamical system to achieve a provided global goal. Poesio *et al.* (2013) use CI from a highly distributed population of contributors to create an internet-scale language resource. The system, called phrase detectives, is developed for distributed information collection. In the big data era, the crowd sensing and crowd sourcing become typical CI applications for information collection in mobile environment (Chen *et al.*, 2014).

On a higher level, Bonabeau (2009) argues that CI can support the decision process but requires a substantial amount of information processing and the evaluation of potential solutions. In this research stream, collaboration modeling and collaboration agents have been used in data processing (Li *et al.*, 2014). Especially, Schut (2010) define that a CI model design methodology is comprised of five steps: assessment, model, simulation, verification, and validation. Modeling is further comprised of generic and specific modeling. Specific model refers to system architecture, and generic model is a formal representation for verification and validation. Among the listed CI system architectures, including self-organization, complex adaptive systems, multi-agent systems, population-based adaptive systems, and swarm intelligence, the multi-agent system is composed of multiple interaction software components, which are typically capable of cooperating to solve problems that are beyond the abilities of any individual member (Wooldridge, 2009). Therefore, its adaptive and proactive features are quite suitable in terms of addressing uncertainty and agile big data system integration problems. For example, Dostatni *et al.* (2013) apply agent technology for recycling-oriented product assessment. Agents help to conduct product design in a distributed environment. In this paper, we use the multi-agent architecture as the specific model to realize the big data integration in a CI framework.

2.3 PN model

A PN is a system modeling and analysis tool. Analyzing a system using a PN model can provide its structural properties and dynamic behavior, which is useful for system improvements, verification, and testing (Murata, 1989; Minns and Elliott, 2008). Moreover, PN models can capture the precedence relations and interactions among events with a strong mathematical foundation, so they have been used in manufacturing systems and supply chain management for a long time (Desrochers and Al-Jaar, 1995; Wu *et al.*, 2007).

The PN literature inspires the big data integration problem in three aspects. First, PN provides data control and data management mechanisms in heterogeneous distributed systems. Simonet *et al.* (2015) have used PN to build a metamodel to describe data life cycle activities. Based on the model, they propose a programming model called "Active Data," to automate and improve data management in Hadoop and MapReduce. Second, PN models have been used to simulate multi-agent systems as a system validation tool. Ma *et al.* (2005) design web services as executive behavior processes of intelligent agents and use a combination of PN models to ensure that users achieve their objectives. Ma and Tsai (2008) extend PN model with object-oriented technologies to describe the secure mobile-agent system. The model supports not only strong mobility, but also secure mechanisms to detect malicious attacks. Third, PNs are a popular tool for QoS simulations in distributed computing. Yang and Ge (2011) propose a QoS-aware and CPN-based web service dynamic composition model (QWS-CPN) and explain some vital elements in the model, such as color aggregations, guard functions, and arc functions. They also introduce a usage-probability-based QoS calculation method to allow service requestors to rapidly obtain the required web services. Nematzadeh *et al.* (2014) develop a deterministic analytical method for dependability and performance measurements using CPNs with explicit routing constructs and probability theory. A tool called WSET is also developed to model and support QoS measurements through simulations.

In this paper, we use PN model as the generic model for system integration in the CI framework. PN describes the data flow interactions and collaborations between agents. System design errors and QoS properties in dynamic environments can be identified during PN simulation.

3. CI-based big data analytics systems

In this section, we utilize the multi-agent paradigm in CI to design integrated big data analytics systems. It has advanced features to fulfill aforementioned requirements as following:

(1) The separation of data and behavior.

In a big data environment, both data storage and processing is substantially more complex than in a database system. Our design separates data management and business processes to fulfill various data storage and data manipulation requirements.

(2) Flexible process composition.

As stated in Section 1, a big data analytics model evolves with the changing environment. Therefore, a flexible system architecture ensures the automatic composition of different functional modules.

(3) Automatic resource dispatching.

A system typically cannot predict the amount of data contained in a big data environment. Thus, we employ the self-creation and proactiveness of agents to realize the load balance and ensure a minimal response time.

3.1 Multi-agent-based system architecture

According to Miller and Mork (2013), big data analytics means a value chain from data to decisions through a series of processes, including data discovery, data integration, and data exploitation. These data processes are not new to software systems. However, how to integrate them to support the aforementioned features in a big data analytics environment is a new challenge. In this research, we choose the multi-agent paradigm. The key to the design is to separate data from behavior. Each behavior is addressed by a group of agents. The data and data-transfer contracts then become the primary organizing constructs. With controlled data relations and timing, the system can then be built from independent agents with loosely coupled behaviors. This data-driven design technique is naturally supported by the data distribution service specification, which is a standard from the object management group. The system architecture is shown in Figure 1. Triangles represent intelligent agents. The solid triangles are the administrators in a group of agents. Administrator agents attempt to create and manage other agents involved in a certain task. A group of agents that share the same goal base is called an agent platform.

The data management layer provides the basic process functions for various types of data. Different agent platforms perform different actions on SQL-like, non-SQL-like or memory-based data for storage, access, and integration. Because big data systems typically require real-time functionality, such as in stream computing technology, we design several administrator agents to perform such no-storage data processes combined with traditional database operations in distributed agent platforms.

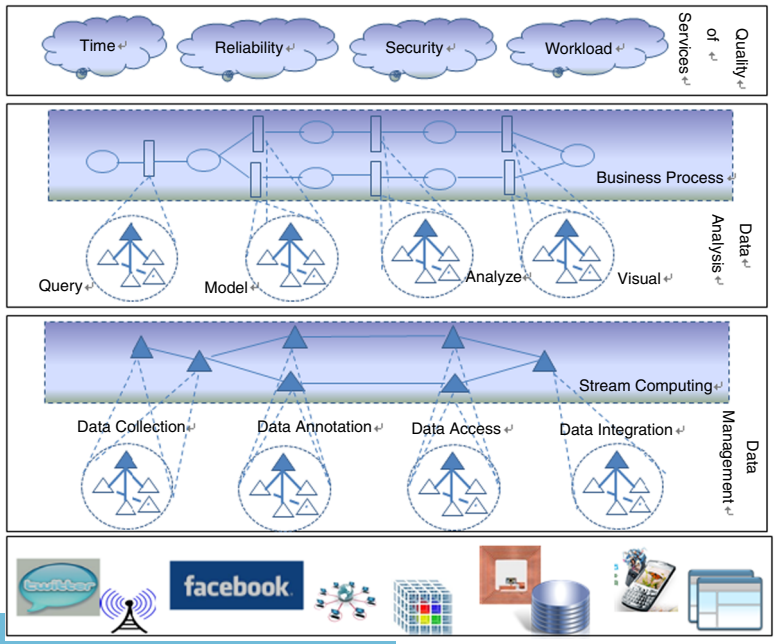

Figure 1.
Framework for the multi-agent-based big data analytics system

The processes in stream computing would be rapidly organized by the multi-agent mechanism as soon as the data are provided.

The data analysis layer attempts to analyze data for data exploitation and decision making. The main actions on the data include query, model, analyze, and visualize. Each action can be supported by a group of agents with their goal and knowledge base. These agents represent existing software systems, web services, cloud applications, or any other participants in an organization. A composition of the four types of agents can be used in various business applications.

The QoS layer provides information exchange contracts between agents. Traditional messaging designs focus on functional or operational interfaces. However, in the multi-agent system, the interface specifies the common, logically shared data model that are produced and used by an agent along with the QoS requirements as its goal, including timing, reliability, workload, and security. With such explicit QoS terms, responses to impedance mismatches can be automated, monitored, and governed.

3.2 Agent communications

According to the functions of an agent, the system contains administrator agents and executive agents. An administrator agent attempts to receive a task, make an executive plan, create executive agents, control the QoS, and send orders. The agent has a goal set and a group of calculation functions to make a plan. An executive agent attempts to realize a specific task according to orders from an administrator agent.

Communication between an administrator agent and executive agent. An administrator agent sends various orders, such as creation, execution, and delete, to executive agents. This agent also sends a data set for processing if necessary. The executive agent sends various statuses, such as finished, failed, and interrupted, to the administrator agent. This agent also sends the processed data back to the administrator agent in the data part.

Communication between executive agents. If a task requires the collaboration of agents, executive agents can communicate with each other through statuses and data.

Communication between administrator agents. The interactions between administrator agents aim at passing data and tasks between them. Figure 2 shows a detailed agent communication.

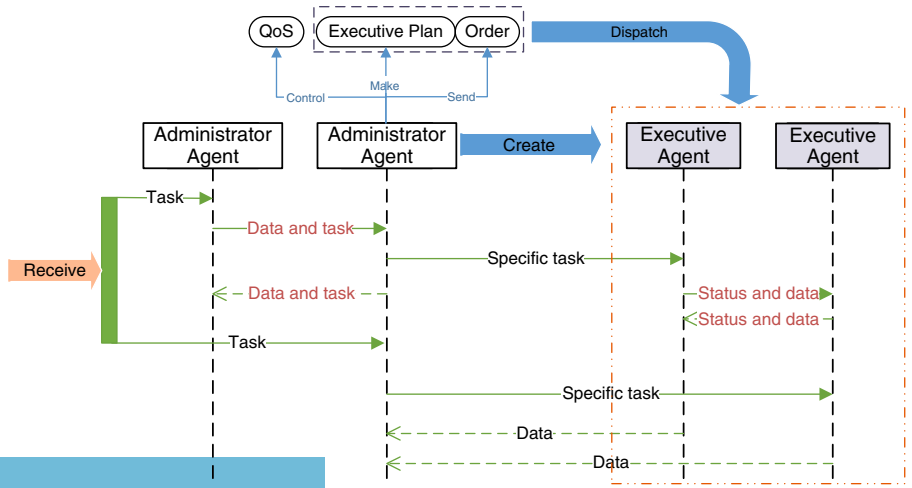

Figure 2.
Agent communications in a CI system

4. A PN model for big data system verification

In system design, it is necessary to examine the architecture consolidations under different environments. A PN model is a formal tool used to simulate system operation. In this section, we use the PN model to describe the designed multi-agent architecture in Section 3.

4.1 Definition of the PN model

PNs are state-transition systems that extend a class of nets called elementary nets (Reisig *et al.*, 1998). CPNs attach a color to each token and a color set to each place and allow us to use fewer places than would be needed in a PN without losing any properties (Xu and Shatz, 2001). The basic idea behind HCPNs is to allow the modeler to construct a large model by combining a number of small CPNs into a larger net. Their formal definitions are listed below:

Definition 1. A CPN is a seven tuple CPN = (Σ,*P*,*T*,*F*,*E*,*C*,*G*), where:

- Σ is a finite and non-empty color set, Σ = {QoS, External Communication, Internal Communication}, where QoS represents information about the QoS, External Communication is the communications between an agent and its external environment, Internal Communication means communications occurring between agents;
- $P = \{p_1, p_2, \ldots p_n\}$ is a finite and non-empty set of places;
- $T = \{t_1, t_2, \ldots t_n\}$ is a finite and non-empty set of transitions, $P \cap T = \varnothing$;
- F is a finite set of arcs, $F \in (P \times T) \cup (T \times P)$;
- E is a function defined on F, E: $F \rightarrow$ Boolexpression, in which $\forall f \in F$: Type($E(f)$) = $C(p)_{MS} \wedge$ Type(Var($E(f)$) $\subseteq \Sigma$, p is a place on f, and $C(p)_{MS}$ is the color on p;
- C is a color function that assigns a color to each place, C: $P \rightarrow \Sigma$, $C(p)$ is from Σ; and
- G is a guard function, G: $T \rightarrow$ Boolexpression, in which $\forall t \in T$: Type(Var($G(t)$)) $\subseteq \Sigma \wedge$ (Type($G(t)$) = Boolean.

Definition 2. A HCPN is defined as a four tuple HCPN = (CPN, *Ts*, *ST*, *PA*).

- CPN is a colored Petri Net.
- *Ts* is a transition set, $\forall t \in T_s$: t = (HCPN′, P_p,*PT*). HCPN is the super page and HCPN' is the subpage. P_p is a set of portal places connecting the super page and subpage. *PT* is a function on P_p, defined as $P_p \rightarrow \{in, out, io\}$, where *in* is the input place, *out* is the output place, and *io* is the input and output place.
- *ST* is a group of functions that map places and transitions to $\{in, out, io\}$, $\forall t \in T_s$:

$$ST(p,t) = \begin{cases} in\ p \in t - t \\ out\ p \in t - t \\ io\ p \in t \cap t \end{cases}$$

- *PA* is a portal place assignment function, which connects super pages and subpages.

An intelligent big data analytics system is modeled by a two-level hierarchical PN model. The upper layer is called the system layer and describes an abstract and functional process. In this layer, transactions model the basic structures of data processes. Resources, such as data and knowledge, are modeled using places. The detailed structures and behaviors of agents are described in the multi-agent platform layer. In this layer, behaviors in intelligent agents are modeled by transitions, and places model the holders of agents and agent execution-supporting information bases (e.g. knowledge bases and goal bases). Every transaction in the system layer is substituted and could be replaced by the model in the multi-agent platform layer. In other words, each data processing action with its input and output could be realized by a group of proactive agents. Therefore, a system can change the structure or behavior of some part without changing its outside world if the in/out ports remain the same. This characteristic grants the model a substantial amount of flexibility and extendibility.

The PN model of an agent platform is shown in Figure 3 and Table II. Its components are illustrated as follows.

Internal communication. Communications among agents on a platform or between platforms are described as internal communications in the model. A generic communication message has the properties of from, end, data, and QoS. From and end control the data flow, and QoS defines the processing requirements. Specifically, we consider the workload and response time as two representative measurements of QoS in our paper. They are defined as a color set in a CPN.

QoS is critical in big data applications. QoS reflects the system performance and forms the basis for system tuning in the design and simulation stage. In terms of big data workload, if the data size is large, the system requires many agents to work corporately; thus, QoS will be used to test traffic delays and data discards during simulation. In terms of big data analysis response time, the QoS ensures that the system operational time is within a reasonable range.

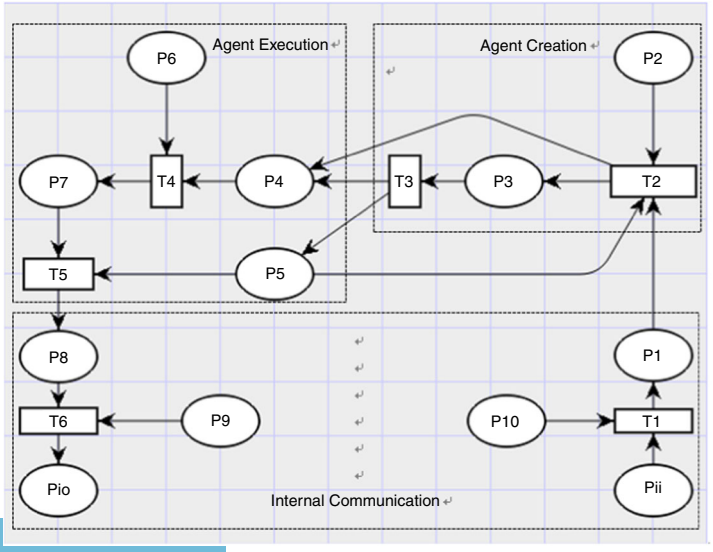

Figure 3.
Petri Net model for a multi-agent platform

Places	Meaning	Transitions	Functions
Pii	Internally incoming communication channels	T1	Process internally incoming information
Pio	Internally outgoing communication channels	T2	Create agent
P1	Internal incoming communication information holder	T3	Maintain an agent list
P2	Goal base	T4	Execute agents
P3	Holder for self-created agents	T5	Integrate results
P4	Holder for agents ready for execution	T6	Process internally outgoing information
P5	Record agents in this platform		
P6	Knowledge base		
P7	Holder for agents after execution		
P8	Internal ingoing communication information holder		
P9/P10	Input/output constraints		

Table II.
Legend for Figure 3

Agent creation. The self-creation of intelligent agents is an important behavior and enables the flexibility and extendibility of the system. In the model, when an administrator agent (T2 in Figure 3) receives an internal communication, it will extract the QoS information and generate a group of agents with its goal base (P2 in Figure 3). For example, for a given workload of 1,000, if the goal base sets the maximum workload for each agent as 100, the administrator will likely generate ten agents for the task.

Agent execution. Once an agent is created, it is placed in the agent list for execution. All agents run in a parallel manner with the support of the knowledge base (P6 in Figure 3). Their results are merged in T5 with a record of sub tasks for each agent (P5 in Figure 3).

4.2 Simulation and evaluation

Simulations and simulation-based analyses of the PN provide a thorough review of system behaviors and characteristics. Many tools support such simulation and analysis. With the CPN meta language (CPN_ML), we can write various codes to simulate agent behaviors. For example, a time delay is assigned to each transaction to test the timing property of the model. Regarding the data type declaration, the compound color-set declarations, which include index, list, record, product, and union, are used. For example, we use the following declarations for internal and external communications.

```
Color QoS = product time×security×reliability×workload;
Color external_comm = union Data: data+QoS: qos;
Color internal_comm = union Platid: from_platform_id+Platid: to_platform_id+Data: data+QoS: qos.
```

The CPN provides the mechanisms for the simulation, statistics, and verification of the design. The analysis of the features of proposed multi-agent-based system is summarized in Table III.

5. Case study

We reconsider the requirements from the abovementioned e-commerce company. A common business process in the company is to manage orders, which typically involves order management, customer management, logistics management, and

Type of analysis	Tasks	Representative analysis issues
Design	To verify the big data analysis mechanism	1. Termination. Can the data in the process flow to a terminal state(s)? What is the dead marking? 2. The statistics of the model. How many arcs and nodes are generated to address different data?
Operation	To simulate the operation of business processes	1. The flexibility of the model. When the business process changes, how does the model change?
Performance	To evaluate the execution correctness and performance of the system	1. Correctness. Does the multi-agent system perform correctly? 2. QoS. Does the system attempt to satisfy the QoS requirements?

Table III.
Representative Petri Net-based analysis

storage management modules. We assume that the four parts are independent software systems. Now, the company wants to develop an integrated system to process various dynamic data to support its business. Shall we rebuild every existing software system using big data technologies? No. We only need to develop the integrated big data management and analysis platforms as supporting facilities to provide preliminary analysis results to the business process.

In Figure 4, we describe the system-level PN model. Each transition can be replaced by an agent platform model implemented by a group of agents, as shown in Figure 3. At this level, a big data analytics system is comprised of four components: data management; data analysis; business processes; and external communications. The data management component is used to provide several solutions to store and retrieve

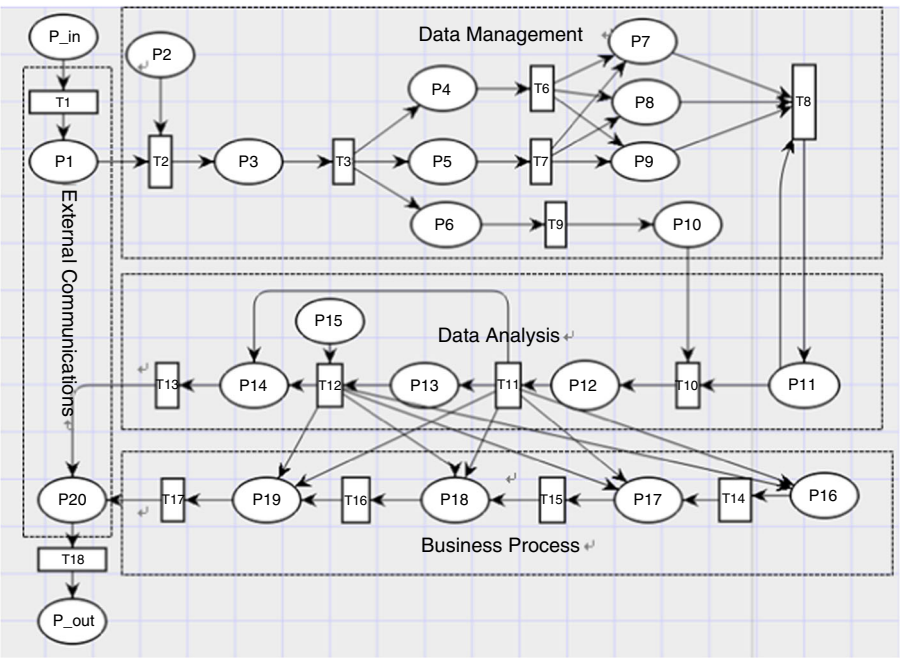

Figure 4.
System-level model

data. The data analysis is used to model and perform a statistical analysis on the queried data. The business process is used to organize the analyzed data in a dynamic flow (Zhao *et al.*, 2014). External communications are used to exchange data with other existing software in the company (Table IV).

In the simulation, we first run the system-level PN model on different components to exam the system design and operation features (see Table V). From the results, we observe that the following: first, all models correctly reach the terminal states because they reach a dead marking status. Second, the complexity of both big data management and big data analytics is greatly reduced compared to their combination with the business process. For example, the data analysis component has 11 nodes and 13 arcs, but there are 3,689 nodes and 14,162 arcs when the data analysis is combined with the business process. This is because most business activities rely on the complex composition of data management and data analytics. Therefore, different business processes may lead to different performances. Third, specifically, a change in business process, especially adding or removing such big data analysis-related activities, may significantly alter the complexity of the entire model. For example, the entire model includes 29,523 nodes and 135,455 arcs. If we delete T14, which is order management, the nodes and arcs are reduced to 5,819 and 22,805, respectively.

Places	Meaning	Transitions	Functions
P1/P20	In/out communication channel	T1/T18	Process in/out communications
P2	Metadata base	T2	Data annotate
P3	Annotated data set	T3/T4/T5	Data store/cache
P4/P5/P6	Date warehouse/Hadoop/memory	T6/T7	Data access
P7/P8/P9	Customer data/order data/storage data	T8	Data integration
P10	Processed data in memory	T9	Stream computing
P11	Integrated data set	T10	Query
P12	Selected data set	T11	Model
P13	Model results	T12	Analyze
P14	Analyzed results	T13	Visualize
P15	Knowledge base	T14	Order management system
P16/P17/P18/P19	Processed order/storage/logistics/ customer data	T15	Storage management system
		T16	Logistics management system
		T17	Customer management system

Table IV. Legend for the system model in Figure 4

Model	State space Nodes	Arcs	Secs	Status	Home marking	Dead marking
Data management	27	49	0	Full	27	27
Data analysis	11	13	0	Full	11	11
Data management and business process	9,498	36,912	12	Full	9,498	9,498
Data analysis and business process	3,689	14,162	3	Full	3,689	3,689
Whole model	29,523	13,5455	224	Full	None	29,514, 29,521, 29,523
Delete T14	5,819	22,805	9	Full	None	5,815, 5,817, 5,819
Delete T14 and T15	1,243	4,037	1	Full	None	1,239, 1,241, 1,243

Table V. Statistics of the Petri Net model at the system level

When performing the agent platform simulation, we measure the agent number and model response time. In the model, we assume that the number of created agents is related to the workload. The maximum workload for each agent is 100. Table VI illustrates that when the workload changes from 100 to 100,000, the number of working agents changes from 1 to 1,000. In a dynamic big data environment, the agent creation mechanism reduces the idle resources during operation to process additional data. In contrast, when the number of created agents increases, the platform execution time also increases. However, the growth rate decreases as the number of agents increases. This shows that such a mechanism can reduce the execution time compared with a traditional pre-designed architecture. Moreover, the simulation can provide an expected response time based on the required workload. This is very useful in an integrated platform design.

6. Discussion and conclusions

According to the simulation results in the case study, it is easily found that, first, multi-agent paradigm is a good implementation for reuse and integration of big data analytics modules in an agile and loosely coupled method. Second, the PN models provide effective simulation results in the system design period. It gives advices on business process design and workload balance control. Third, the CI framework provides an incrementally build and deployed method for system integration. It is especially suitable to the dynamic data analytics environment. These findings have both theoretical and managerial implications.

6.1 Theoretical implication

The proposed integration model first contributes to both big data management and system engineering literatures. In the big data field, although substantial technical progress has been made, a comprehensive theoretical foundation is still needed to drive the design and integration of substantial numbers of participants in the system. In this paper, we propose a CI framework, which includes both practical architectures and theoretical foundations, to solve the system integration problem in big data environment. It provides a new point of view to dynamically integrate large-scale modules in an organization. In the system engineering field, we propose that both generic and specific models are needed during system design. The architecture verification and validation in a formal format is quite useful to evaluate the system design in different situations.

Table VI.
QoS of agent platform simulation

Workload	Agent no.	Time (CPN time unit)
100	1	96
200	2	158
300	3	213
400	4	269
500	5	292
600	6	329
700	7	367
800	8	401
900	9	458
1,000	10	519
10,000	100	4,161
100,000	1,000	40,315

On a higher level, this research contributes to the reuse of knowledge and building the knowledge economy in an organization. In the big data era, decision making usually integrates multiple data analysis outputs. These outputted analysis results constitute knowledge within an organization and display two features. One is the reuse of knowledge, and the other is the loosely integration of knowledge. In this paper, we propose to use CI framework in system integration. The framework, at the same time, also works for knowledge management to generate, organize, reuse, and integrate knowledge by collaborative agents in consensus decision making. This dynamic knowledge creation and consumption pattern would be great valuable for the development of knowledge economy in the big data environment.

6.2 Managerial implication

This paper also has practical suggestions for Chief Technical Officers, who want to employ big data technologies in their companies. There are four steps to follow according to our findings. First, it is useful to wrap up functional big data analytics modules as agents or other reusable components in an organization. Second, it is important to optimize the business process for a task, because different processes may lead to quite different performances. Third, system evaluation is the next step to test the architecture design in different situations. Finally, the whole system is developed in an incremental method with iteration and repetition of previous steps.

6.3 Limitations and future directions

Nevertheless, this study has several limitations that serve as avenues for future research. First, we use PN model for simulations in the case study. The feasibility of multi-agent architecture is not tested. A prototype system would be developed in the future to investigate the effectiveness of using agents in big data analytics modules. Second, we use multi-agent architecture and HCPN in the proposed CI framework. In the future, we want to investigate other CI paradigms and formal verification methods in the general CI framework with different applications.

References

Berkovich, S. and Liao, D. (2012), "On clusterization of 'big data' streams", *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, Article No. 26, *New York, NY, July 1-3*.

Bonabeau, E. (2009), "Decisions 2.0: the power of collective intelligence", *MIT Sloan Management Review*, Vol. 50 No. 2, pp. 45-52.

Chen, H., Chiang, R.H.L. and Storey, V.C. (2012), "Business intelligence and analytics: from big data to big impact", *MIS Quarterly*, Vol. 36 No. 4, pp. 1165-1188.

Chen, M., Mao, S. and Liu, Y. (2014), "Big data: a survy", *Mobile Network Application*, Vol. 19 No. 2, pp. 171-209.

Dalal, N.P., Kamath, M., Kolarik, W.J. and Sivaraman, E. (2004), "Toward an integrated framework for modeling enterprise processes", *Communications of the ACM*, Vol. 47 No. 3, pp. 83-87.

Demirkan, H. and Delen, D. (2013), "Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud", *Decision Support Systems*, Vol. 55 No. 1, pp. 412-421.

Desrochers, A. and Al-Jaar, R. (1995), *Applications of Petri Nets in Manufacturing Systems: Modeling Control and Performance Analysis*, IEEE Press, New York, NY.

Dong, X.L. and Srivastava, D. (2013), "Big data integration", *IEEE 29th Internation Conference on Data Engineering, Brisbane*, pp. 1245-1248.

Dostatni, E., Diakun, J., Hamrol, A. and Mazur, W. (2013), "Application of agent technology for recycling-oriented product assessment", *Industrial Management & Data Systems*, Vol. 113 No. 6, pp. 817-839.

Fernandez, R.C., Pietzuch, P., Koshy, J., Kreps, J., Lin, D., Narkhede, N., Rao, J., Riccomini, C. and Wang, G. (2015), "Liquid: unifying nearline and offline big data integration", *7th Biennial Conference on Innovative Data Systems Research, Monterey, Asilomar, CA, January 4-7.*

Glenn, J.C. (2015), "Collective intelligence systems and an application by the millennium project for the egyptian academy of scientific research and technology", *Technological Forecasting and Social Change*, Vol. 97, August, pp. 7-14.

Jacobs, A. (2009), "The pathologies of big data", *Communications of The ACM*, Vol. 52 No. 8, pp. 36-44.

Khazaei, T. and Lu, X. (2014), "Collective intelligence in massive online dialogues", Cornell University Library, available at: http://arxiv.org/abs/1406.7561 (accessed June 29, 2014).

Knoblock, C.A. and Szekely, P. (2015), "Exploiting semantics for big data intergration", *AI Magazine*, Vol. 36 No. 1, pp. 25-38.

Kounev, S. (2005), "Performance modeling and evaluation of distributed component-based systems using queuing petri nets", *IEEE Transactions on Software Engineering*, Vol. 32 No. 7, pp. 486-502.

Li, Q., Wang, Z., Cao, Z., Du, R. and Luo, H. (2014), "Process and data fragmentation-oriented enterprise network integration with collaboration modelling and collaboration agents", *Enterprise Information Systems*, Vol. 9 Nos 5-6, pp. 468-498. doi: 10.1080/17517575.2013.879742.

Liu, Q. and Lv, W. (2015), "Multi-component manufacturing system maintenance scheduling based on degradation information using generic algorithm", *Industrial Management & Data Systmes*, Vol. 115 No. 8, pp. 1412-1434.

Ma, B.X., Wu, Z.H. and Xie, N.F. (2005), "Modeling agent-based Semantic web services with Petri Nets", *Journal of System Simulation*, Vol. 17 No. 1, pp. 120-123.

Ma, L. and Tsai, J.P. (2008), "Formal modeling and analysis of a secure mobile-agent system", *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, Vol. 38 No. 1, pp. 180-196.

Madden, S. (2012), "From databases to big data", *IEEE, Intenet Computing*, Vol. 16 No. 3, pp. 4-6.

Marx, V. (2013), "Biology: the big challenges of big data", *Nature*, Vol. 498 No. 7453, pp. 255-260.

Miller, H.G. and Mork, P. (2013), "From data to decisions: a value chain for big data", *Computer*, Vol. 15 No. 1, pp. 57-59.

Minns, P. and Elliott, I. (2008), *FSM-Based Digital Design Using Verilog HDL: Introduction to Petri Nets*, John Wiley & Sons Inc., London. doi: 10.1002/9780470987629.ch10.

Murata, T. (1989), "Petri Nets: properties, analysis and applications", *Proceedings of the IEEE*, Vol. 77 No. 4, pp. 541-580.

Nematzadeh, H., Motameni, H., Mohamad, R. and Nematzadeh, Z. (2014), "QoS measurement of workflow-based web service compositions using colored Petri Net", *The Scientific World Journal*, Vol. 2014, Article ID 847930, 14pp. doi: 10.1155/2014/847930.

O'Leary, D.E. (2013), "Artificial intelligence and big data", *IEEE Intelligent Systems*, Vol. 28 No. 2, pp. 96-99.

Poesio, M., Chamberlain, J., Kruschwitz, U., Robaldo, L. and Ducceschi, L. (2013), "Phrase detectives: utilizing collective intelligence for internet-scale language resource creation", *ACM Transactions on Interactive Intelligent Systems*, Vol. 3 No. 1, Article no. 3.

Raden, N. (2012), "Big data analytics architecture", technique report, Hired Brains, Inc., available at: www.bigdatainsightgroup.com/site/sites/default/files/Teradata's%20-%20Big%20Data%20Architecture%20-%20Putting%20all%20your%20eggs%20in%20one%20basket.pdf (accessed July 29, 2015).

Reisig, W., Rozenberg, G., Science, S. and Media, B. (1998), *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, Springer, Verlag Berlin Heidelberg.

Schut, M.C. (2010), "On model design for simulation of collective intelligence", *Information Science*, Vol. 180 No. 1, pp. 132-155.

Simonet, A., Fedak, G. and Ripeanu, M. (2015), "Active data: a programming model to manage data life cycle across heterogeneous systems and infrastructures", *Future Generation Computer Systems*, Vol. 53, December, pp. 25-42.

Talia, D. (2013), "Clouds for scalable big data analytics", *IEEE Computer*, Vol. 46 No. 5, pp. 98-101.

Wang, H.Q., Mylopoulos, J. and Liao, S. (2002), "Intelligent agents and financial risk monitoring systems", *Communications of the ACM*, Vol. 45 No. 3, pp. 83-88.

Wolpert, D.H., Wheeler, K.R. and Tumer, K. (2000), "Collective intelligence for control of distributed dynamical systems", *Europhysics Letters*, Vol. 49 No. 6, pp. 708-714.

Wooldridge, M.J. (2009), *An Introduction to Multi-Agnet Systems*, John Wiley & Sons, BaffinsLane, Chichester.

Wu, T., Blackhurst, J. and O'grady, P. (2007), "Methodology for supply chain disruption analysis", *International Journal of Production Research*, Vol. 45 No. 7, pp. 1665-1682.

Xu, H. and Shatz, S.M. (2001), "An agent-based petri net model with application to seller/buyer design in electronic commerce", *Proceedings of 5th International Symposium on Autonomous Decentralized Systems IEEE, Dallas, TX*, pp. 11-18.

Yang, L.Q. and Ge, X.K. (2011), "Study on a QoS-aware and color petri net based webs service dynamic composition", *Computer Applications and Software*, Vol. 28 No. 5, pp. 102-105.

Zhao, S., Wang, L. and Zheng, Y. (2014), "Integrating production planning and maintenance: an iterative method", *Industrial Management & Data Systems*, Vol. 144 No. 2, pp. 162-182.

About the authors

Kun Chen is an Assistant Professor in the Department of Financial Mathematics and Financial Engineering at the South University of Science and Technology of China. She received her PhD from the Department of Information Systems at the City University of Hong Kong. Her research interests include big data analytics, financial intelligence, and social network analysis. Kun Chen is the corresponding author and can be contacted at: chenk@sustc.edu.cn

Xin Li is an Assistant Professor in the Department of Information Systems at the City University of Hong Kong. He received his PhD in Management Information Systems from the University of Arizona. He received his bachelor's and master's degrees from the Department of Automation at the Tsinghua University, China. His research interests include business intelligence and knowledge discovery, social network analysis, social media, and e-commerce.

Huaiqing Wang is a Professor in the Department of Financial Mathematics and Financial Engineering at the South University of Science and Technology of China. He is also the Honorary Dean and a Guest Professor of the School of Information Engineering, Wuhan University of Technology, China. He received his PhD from the University of Manchester, UK, in 1987. Dr Wang specializes in research of Financial Intelligence, and Intelligent Systems (such as intelligent financial systems, intelligent learning systems, business process management systems, knowledge management systems, conceptual modeling, and ontology). He has published more than 70 international refereed SCI/SSCI journal articles and received more than 700 SCI citations.