

명세서

발명의 명칭: 스크래핑 기반 서비스 제공 방법 및 이를 수행하는 어플리케이션

기술분야

- [1] 본 발명은 스크래핑 기반 서비스 제공 방법에 관한 것이다.

배경기술

- [2] 스크래핑은 각각 분산된 데이터들 중에서 정보 제공에 필요한 데이터를 추출해 사용자에게 제공하는 기술로, 예를 들어 금융사나 공공기관, 정부 웹사이트 등 여러 곳에 흩어져 있는 정보를 고객의 인증 정보를 이용하여 통합하여 조회하고 관리할 수 있도록 한다.
- [3] 이 중 스크린 스크래핑은 스크래핑 중의 한 방식으로 디스플레이 장치를 통해 화면에 보여지는 데이터를 수집하여 제공하는 것으로, 일반적으로 웹사이트 등에 표시되는 정보를 수집하여 제공하므로 웹 스크래핑이라고도 한다.
- [4] 주로 각각의 금융기관에 흩어져 있는 개인의 자산이나 지출을 관리하는 분야에서 이용되며 서비스 마일리지를 통합 관리하거나, 이메일을 통합 조회하여 사용자에게 제공하기 위해 스크래핑 기술들이 이용되고 있다.
- [5] 즉, 스크래핑을 통하면 각각의 정보 관리 주체로 분산된 정보를 한데 모아 관리할 수 있는 장점이 있다.
- [6] 스크래핑을 통한 서비스를 제공하는 주체는 스크래핑 대상 정보가 관리되는 웹사이트 내 인증 방식이나 시스템 요구 사항의 변경에 따라 개별적이고 지속적인 대응이 필요하다.
- [7] 종래의 스크래핑을 통해 서비스를 제공하는 어플리케이션은 대상 사이트의 변경에 직접 대응하기 위해 대상 웹 사이트 별로 별도의 스크래핑 모듈을 구성하고 있으며, 변경 사항을 지속적으로 모니터링 하고 스크래핑 모듈을 대상 사이트의 변경에 따라 업데이트하는 방식으로 운영되고 있다.
- [8] 이러한 스크래핑 모듈의 변경은 하나의 서비스를 제공하기 위해 복수의 스크래핑 대상 웹 사이트가 존재하는 경우, 각각의 웹 사이트의 변경이 있을 때마다 어플리케이션을 수정하고 재 배포해야 하는 문제를 초래하게 된다.
- [9] 사용자 입장에서는 서비스 이용을 위해서 설치된 어플리케이션을 매번 재설치해야 하는 문제가 발생하므로 사용에 불편함을 느끼게 된다.
- [10] 따라서, 이러한 문제를 해결하기 위하여 스크래핑 대상의 변경 상황에 능동적으로 대응하고 어플리케이션의 업데이트를 최소화하여 스크래핑 기반의 서비스를 제공하는 방법의 고안이 필요한 실정이다.

발명의 상세한 설명

기술적 과제

- [11] 상술한 기술적 과제를 해결하기 위해 본 발명은 스크래핑 서비스를 효율적으로

제공하는 방법을 제안하는 것을 목적으로 한다.

- [12] 보다 상세하게는 스크래핑 대상 주체의 변경에 능동적으로 대응하고, 스크래핑을 위한 업무를 보다 효율적으로 수행하는 방법을 제안하는 것을 목적으로 한다.

- [13] 또한, 본 발명은 스크래핑 서비스를 위한 라이브러리를 구성하는 방식을 분산하고, 따라서 별도의 어플리케이션의 업데이트나 재배포를 줄이는 것을 목적으로 한다.

기술적 해결방법

- [14] 상기 기술적 과제를 해결하기 위한 본 발명에 따른 스크래핑 기반 서비스 제공 방법은 외부 데이터를 획득하기 위한 일련의 태스크를 정의하는 스크래핑 룰(Rule) 정보 중 적어도 하나의 룰 정보를 요청하여 수신하는 단계; 상기 수신된 룰 정보에 따른 태스크를 수행하여 상기 외부 데이터를 수집하는 단계; 및 상기 수집된 외부 데이터를 이용하여 서비스 정보를 사용자에게 제공하는 단계를 포함한다.
- [15] 상기 수집하는 단계는, 상기 룰 정보 내 정의된 태스크를 구성하는 복수의 커맨드를 클라이언트 단말 또는 원격 서버 상에서 수행하여 상기 데이터를 수집하는 것이 바람직하다.
- [16] 상기 태스크는 상기 클라이언트 단말 상에서 수행되는 미리 생성된 로컬 커맨드의 조합으로 구성되는 것이 바람직하다.
- [17] 상기 태스크는 상기 원격 서버 상에서 수행되는 맞춤형(Custom)으로 생성된 원격 커맨드를 더 포함하는 것이 바람직하다.
- [18] 상기 원격 커맨드의 수행을 위해 상기 클라이언트는 원격 프로시저 콜(RPC:Remote Procedure Call)을 상기 원격 서버에 송신하고,
- [19] 상기 원격 서버는 상기 원격 프로시저 콜에 따라 수행된 태스크의 수행 결과를 상기 클라이언트 단말에 리턴하는 것이 바람직하다.
- [20] 상기 수집하는 단계는, 상기 수행되는 태스크의 수행 결과를 매 태스크의 수행마다 검증하되,
- [21] 상기 태스크의 수행에 이상이 있는 경우 상기 룰 정보의 수정 또는 상기 커맨드에 대한 원격 프로시저 콜을 요청하는 것이 바람직하다.
- [22] 상기 태스크의 수행 스케줄에 따라 상기 원격 커맨드가 소정 횟수 이상 연속하는 경우,
- [23] 상기 클라이언트 단말은 커맨드 수행을 제어하는 제어권을 상기 원격 서버로 제공하고, 제어권을 수신한 상기 원격 서버에서 연속하여 수행된 원격 커맨드의 수행 결과와 제어권을 리턴 받는 것이 바람직하다.
- [24] 상기 태스크는 블록 단위로 수행되어 상기 태스크내 커맨드의 수행 결과는 상기 블록 내에 저장되고,
- [25] 상기 태스크의 참조 시 태스크 명 및 관련 변수를 이용하여 상기 태스크의 수행

결과를 참조하는 것이 바람직하다.

- [26] 상기 로컬 커맨드 및 상기 원격 커맨드는 상기 클라이언트 단말 또는 원격 서버를 통해 분산 수행되고,
- [27] 상기 태스크는 비동기식 폴링되는 것이 바람직하다.
- [28] 상기 기술적 과제를 해결하기 위한 본 발명에 따른 스크래핑 기반 서비스 제공 어플리케이션은, 외부 데이터를 획득하기 위한 일련의 태스크를 정의하는 스크래핑 룰(Rule) 정보 중 적어도 하나의 룰 정보를 요청하여 수신하는 스크래핑 룰 요청부; 상기 수신된 룰 정보에 따른 태스크를 수행하여 상기 외부 데이터를 수집하는 태스크 수행부; 및 상기 수집된 외부 데이터를 이용하여 서비스 정보를 사용자에게 제공하는 서비스 제공부를 포함하는 것이 바람직하다.
- [29] 상기 태스크 수행부는, 상기 룰 정보 내 정의된 태스크를 구성하는 복수의 커맨드를 클라이언트 단말 또는 원격 서버 상에서 수행하여 상기 데이터를 수집하는 것이 바람직하다.
- [30] 상기 태스크 수행부는, 상기 원격 커맨드의 수행을 위해 원격 프로시저 콜(RPC:Remote Procedure Call)을 상기 원격 서버에 송신하는 RPC 송신부; 및
- [31] 상기 원격 서버는 상기 원격 프로시저 콜에 따라 수행된 태스크의 수행 결과를 리턴 받는 커맨드 결과 수신부를 더 포함하는 것이 바람직하다.
- [32] 상기 기술적 과제를 해결하기 위한 본 발명에 따른 스크래핑 기반 서비스 원격 제공 원격 방법은, 외부 데이터를 획득하기 위한 일련의 태스크를 정의하는 스크래핑 룰(Rule) 정보를 관리하고, 클라이언트 단말의 요청에 따라 상기 룰 정보를 송신하는 단계; 상기 클라이언트 단말의 상기 송신된 룰 정보 내 정의된 태스크의 수행 중 원격 수행이 필요한 원격 커맨드를 원격 프로시저 콜에 따라 수행하는 단계; 및 상기 원격 커맨드의 수행 결과를 상기 클라이언트 단말로 리턴하는 단계를 포함하는 것이 바람직하다.
- [33] 상기 수행하는 단계는, 상기 태스크의 수행 스케줄에 따라 상기 원격 커맨드가 소정 횟수 이상 연속하는 경우, 상기 클라이언트 단말로부터 커맨드 수행을 제어하는 제어권을 수신하고 상기 원격 커맨드를 연속하여 수행하는 것이 바람직하다.
- [34] 상기 리턴하는 단계는, 상기 연속하여 수행된 원격 커맨드의 수행 결과와 상기 제어권을 수신한 상기 원격 서버에서 연속하여 수행된 원격 커맨드의 수행된 결과와 제어권을 리턴하는 것이 바람직하다.
- [35] 상기 기술적 과제를 해결하기 위한 본 발명에 따른 스크래핑 기반 서비스 제공 원격 서버는, 외부 데이터를 획득하기 위한 일련의 태스크를 정의하는 스크래핑 룰(Rule) 정보를 관리하고, 클라이언트 단말의 요청에 따라 상기 룰 정보를 송신하는 룰 서버; 및 상기 클라이언트 단말의 상기 송신된 룰 정보 내 태스크의 수행 중 원격 수행이 필요한 커맨드를 원격 프로시저 콜에 따라 수행하고, 수행 결과를 리턴하는 커맨드 서버를 포함하는 것이 바람직하다.

[36] 상기 커맨드 서버는 상기 태스크의 수행 스케줄에 따라 상기 원격 커맨드가 소정 횟수 이상 연속하는 경우, 상기 클라이언트 단말로부터 커맨드 수행을 제어하는 제어권을 수신하고 상기 원격 커맨드를 연속하여 수행하는 것이 바람직하다.

[37] 상기 커맨드 서버는, 상기 연속하여 수행된 원격 커맨드의 수행 결과와 상기 제어권을 수신한 상기 원격 서버에서 연속하여 수행된 원격 커맨드의 수행된 결과와 제어권을 리턴하는 것이 바람직하다.

발명의 효과

[38] 본 발명에 따르면, 스크래핑 서비스를 클라이언트 단말과 원격 서버를 통해 분산 제공함으로써 스크래핑 업무의 효율을 높이고, 스크래핑 대상의 변경 사항에 능동적으로 대응할 수 있다.

[39] 또한, 대상의 변경에 대하여 어플리케이션의 업데이트나 재배포를 최소화하여 사용자의 이용 편의성을 높일 수 있다.

[40] 또한, 스크래핑을 수행하는 원격 서버를 클라이언트 단말과 독립적으로 운영하여 보안성을 높이고 스크래핑 수행을 위해 이용되는 정보에 대한 외부의 접근과 유출을 방지할 수 있다.

도면의 간단한 설명

[41] 도 1은 본 발명의 일 실시예에 따른 스크래핑 기반 서비스를 제공하는 시스템 구성을 예시하는 도이다.

[42] 도 2는 본 발명의 일 실시예에 따른 스크래핑을 수행하는 클라이언트 단말과 스크래핑 프로세스를 제공하는 원격 서버로 구성된 시스템을 예시하는 도이다.

[43] 도 3은 본 발명의 일 실시예에 따른 클라이언트 단말과 물 서버 간의 물 정보를 요청하는 프로세스를 예시하는 도이다.

[44] 도 4를 본 발명의 일 실시예에 따른 클라이언트 단말의 태스크 수행 프로세스를 예시하는 도이다.

[45] 도 5는 본 발명의 일 실시예에 따른 클라이언트 단말 상에서 어플리케이션이 스크래핑 서비스를 수행하는 방법을 나타내는 도이다.

[46] 도 6은 본 발명의 일 실시예에 따른 클라이언트 단말의 태스크 수행 프로세스를 예시하는 도이다.

[47] 도 7a 및 7b는 본 발명의 일 실시예에 따른 태스크들의 관계 및 수행 순서를 예시하는 도이다.

[48] 도 8을 본 발명의 일 실시예에 따른 어플리케이션의 구성을 예시하는 도이다.

발명의 실시를 위한 형태

[49] 이하의 내용은 단지 발명의 원리를 예시한다. 그러므로 당업자는 비록 본 명세서에 명확히 설명되거나 도시되지 않았지만 발명의 원리를 구현하고 발명의 개념과 범위에 포함된 다양한 장치를 발명할 수 있는 것이다. 또한, 본 명세서에 열거된 모든 조건부 용어 및 실시 예들은 원칙적으로, 발명의 개념이

이해되도록 하기 위한 목적으로만 명백히 의도되고, 이외같이 특별히 열거된 실시 예들 및 상태들에 제한적이지 않는 것으로 이해되어야 한다.

- [50] 상술한 목적, 특징 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해질 것이며, 그에 따라 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.
- [51] 또한, 발명을 설명함에 있어서 발명과 관련된 공지 기술에 대한 구체적인 설명이 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다. 이하에는 첨부한 도면을 참조하여 본 발명의 바람직한 실시 예에 대해 상세하게 설명한다.
- [52] 도 1은 본 발명의 일 실시예에 따른 스크래핑 기반 서비스를 제공하는 시스템 구성을 예시하는 도이다.
- [53] 본 실시예에서 시스템은 사용자에게 서비스를 직접 제공하는 클라이언트 단말(10)과 스크래핑 대상 페이지를 제공하는 웹 사이트, 웹 서버, 어플리케이션 등의 인터넷 기반의 정보 제공 주체(20)로 구성될 수 있다.
- [54] 클라이언트 단말(10)은 웹 사이트 상에 흩어져 있는 다양한 데이터 중 필요한 데이터를 추출하여 사용자에게 통합 서비스를 제공하기 위한 수단으로서, 스크린 스크래핑(Screen Scraping)을 수행할 수 있다.
- [55] 사용자는 클라이언트 단말(10)에 서비스를 제공하는 어플리케이션(1000)을 설치하고, 어플리케이션(100)은 클라이언트 단말(10)을 통해 원하는 기능을 제공하기 위해 모듈을 라이브러리(100)로 구성하고 원하는 서비스를 사용자에게 제공해 줄 수 있다.
- [56] 본 실시예에서 어플리케이션(1000)은 사용자에게 스크래핑 기반의 특정 서비스를 제공해주기 위한 태스크들을 수행하는 모듈들의 집합을 라이브러리(100)로 구성할 수 있다.
- [57] 라이브러리(100) 내 특정 모듈은 외부의 웹 사이트(20)에 접속하고, 마치 사용자가 직접 웹 사이트(20)의 서비스를 이용하는 것처럼 로그인이나 사용자 공인 인증을 수행하고, 데이터를 획득하는 프로세스를 순차적으로 수행할 수 있다.
- [58] 이하, 도 2를 참조하여 본 실시예에 따른 라이브러리(100)의 태스크 수행 방법에 대하여 보다 상세히 설명한다.
- [59] 도 2는 본 발명의 일 실시예에 따른 스크래핑을 수행하는 클라이언트 단말(10)과 스크래핑 프로세스를 제공하는 원격 서버(50)로 구성된 시스템을 예시하는 도이다.
- [60] 클라이언트 단말(10) 상에 구성되는 라이브러리(100)는 스크래핑을 위한 태스크를 수행하는 태스크 수행부(110)를 포함할 수 있다.
- [61] 태스크 수행부(110)는 인터넷을 통해 외부의 웹 사이트(20)에 접속하여 정보를 송수신 하며 웹 사이트(20)에 포함된 데이터들을 스크래핑하여 수집한다.
- [62] 원격 서버(50)는 클라이언트 단말(10)과 통신을 통해 스크래핑 룰(RULE)

정보들을 제공하고, 원격 수행이 필요한 커맨드들을 원격 프로시저 콜(RPC:Remote Procedure Call)을 통해 직접 수행할 수 있다.

- [63] 룰 정보는 태스크의 종류와 수행 순서를 정의할 수 있다. 예를 들어, A, B, C, D의 태스크가 있을 때, 룰 정보는 각 태스크의 수행 순서를 A-B-C 또는 A-B-D를 정의할 수 있다.
- [64] 또한, 룰 정보 내 태스크는 오류 탐지나 스크래핑의 효율성을 고려하여 구성 또는 재구성 될 수 있다.
- [65] 예를 들어 태스크를 오류 메시지 단위로 구분하여 구성하면, 스크래핑 과정에서 오류 메시지가 수신되면 어떠한 태스크로 인한 오류 인지 원인 파악이 쉬워질 수 있다.
- [66] 나아가, 이러한 태스크를 계층화하여 관리하는 것도 가능하다.
- [67] 상술한 바와 같이 태스크를 오류 메시지 단위로 구분하게 되면 오류 탐지 측면에서는 효율적일 수 있으나 통신 측면에서 비효율성을 초래할 수 있다. 예를 들어, 스크래핑의 특정 상 태스크 수행 과정에서 외부 서버와 데이터의 송수신이 발생하게 되는데 매 송수신 결과를 메시지로 리턴 받는 것은 비효율적일 수 있다.
- [68] 즉, 태스크의 상세한 구분이 전체 스크래핑의 효율성을 저해할 수 있으므로 일부 태스크를 하나의 상위 태스크로 그룹화하여 관리할 수 있다. 예를 들어 순차적인 태스크를 정해진 개수로 묶어서 하나의 상위 태스크로 관리하고, 상위 태스크를 단위로 데이터 송수신 및 이에 따른 결과를 리턴하도록 할 수 있다.
- [69] 이때 상위 태스크에 속한 하위 태스크 하나에서 수행 중 오류가 발생하면 상위 태스크를 단위로 오류 메시지가 생성될 수 있다. 따라서 관리자는 상위 태스크에 속한 하위 태스크에 대해서만 구체적인 오류 여부를 판단하도록 하여 효과적으로 태스크의 수행 및 관리가 가능하도록 할 수 있다.
- [70]
- [71] 원격 서버(50)는 룰 서버(52), 커맨드 서버(54) 및 컨텍스트 서버(58)를 포함하여 구성될 수 있다.
- [72] 룰 서버(52)는 스크래핑을 위한 태스크 및 태스크의 시퀀스를 미리 룰 정보로 정의하고, 정의된 룰 정보를 저장 및 관리하는 장치로 원격으로 접속 가능할 수 있다. 따라서, 관리자는 룰 서버(52)에 접속하여 정의된 룰 정보를 직접 수정할 수 있으며 스크래핑 대상의 변경에 따라 달라지는 태스크를 룰 정보에 직접 반영할 수 있다.
- [73]
- [74] *커맨드 서버(54)는 클라이언트 단말(10)과 연동하여 원격으로 특정 커맨드를 수행할 수 있다. 룰 서버(52)에서 제공된 룰 정보에 따라 스크래핑을 수행하되, 원격으로 수행이 필요한 커맨드에 대해서 직접 수행을 요청 받고 수행 결과를 클라이언트 단말(10)에 리턴 할 수 있다.
- [75] 즉, 커맨드는 태스크에 포함되는 명령어로 저장 위치에 따라 클라이언트

단말(10)에 저장되는 로컬 커맨드와 커맨드 서버(54)에 저장되는 원격 커맨드로 구분될 수 있다.

- [76] 클라이언트 단말(10)에 저장된 로컬 커맨드는 어플리케이션(1000)에 포함되어 설치된 것으로, 라이브러리(100) 내에 구성될 수 있다. 따라서 일반적으로 클라이언트 단말(10)에서 수정이 불가능한 정적 형태를 가질 수 있다.
- [77] 반면, 커맨드 서버(54)에 저장되는 원격 커맨드는 관리자가 커맨드 서버(54)에 접속하여 자유롭게 수정 가능한 것으로, 스크래핑의 목적에 맞게 커스터마이징(customizing)되거나 수정이 가능한 동적 형태를 가질 수 있다.
- [78] 또한, 커맨드 중 일부는 테스트 커맨드로 구성될 수 있다.
- [79] 테스트 커맨드는 사용자의 요청이 없더라도 주기적으로 수행될 수 있으며, 실제 요청에 의한 스크래핑 업무를 수행하는 것과 같은 내용으로 구성될 수 있다.
- [80] 따라서, 테스트 커맨드는 주기적으로 실행되며, 주기적으로 스크래핑 대상의 변경을 탐지하고 이에 따른 오류 발생을 판단할 수 있다.
- [81]
- [82] 이상의 룰 서버(52)와 커맨드 서버(54)는 원격으로 관리되며 서로 독립하여 구성될 수 있다. 커맨드 서버(54)에서 실제 스크래핑을 위해 수행되는 프로세스는 외부에서 접근이 불가하고 오직 룰 서버(52)에 정의된 태스크에 의해서만 제어될 수 있다.
- [83] 따라서 원격으로 실행되는 커맨드에 이용되는 정보들의 유출을 방지할 수 있으며 예를 들어 사용자의 공인 인증 정보나 로그인에 필요한 암호 등의 유출을 방지할 수 있다.
- [84] 다만, 본 실시예에서 룰 서버(52)와 커맨드 서버(54)는 물리적으로 별개의 주체로 구분되어 동작하는 것 외에 원격 서버(50) 내에 구성되어 함께 관리되는 것도 가능하다.
- [85] 일 예로, 원격 서버(50) 내의 장치로서 서로 다른 업무를 수행하는 내부 모듈로 구성되는 것도 가능하다. 원격 서버(50) 내에 룰 모듈과 커맨드 모듈로 각각 구성되어 동작하되 커맨드 모듈의 경우는 외부로부터 제어나 접근이 불가능하도록 원격 서버 내에 구현될 수 있다.
- [86] 컨텍스트 서버(58)는 커맨드 서버(54)에서 태스크 수행 시에 참조할 값을 저장할 수 있다.
- [87] 예를 들어 컨텍스트 서버(58)는 태스크명(A, B, C)을 포함한 컨텍스트 데이터(A.Block, B.Title, C.Desc)를 구성하며, 이때 컨텍스트 데이터에는 실제 태스크 수행 시에 참조할 데이터 값 등의 상세 내용들이 포함될 수 있다.
- [88] 컨텍스트 서버(58)는 커맨드의 실행 결과를 컨텍스트 데이터로 다시 저장하게 되는데, 이때 해당 커맨드가 포함된 태스크명을 붙여서 저장할 수 있다. 따라서 실제 저장 시 스크래핑 세션별로 구분된 컨텍스트 데이터로 저장될 수 있다.
- [89] 컨텍스트 데이터는 "태스크명.변수명(A.Block)"과 같은 형태로 저장되어

다양하게 정의되는 변수의 충돌 가능성을 줄임과 동시에 태스크들 간의 의존 관계를 파악할 수 있도록 할 수 있다.

- [90] 예를 들어, 관리자는 참조되는 컨텍스트 데이터 A.Block으로부터 해당 컨텍스트 데이터는 A 태스크의 결과임을 쉽게 파악할 수 있으며, B 태스크 중 일부 커맨드가 A.Block를 참조하는 것으로 B 태스크의 A 태스크에 대한 의존 관계를 파악할 수 있다.
- [91] 또한, 의존 관계를 통하여 태스크 내 커맨드의 상호 참조로 무한 루프가 발생하는 지 및 이로 인한 태스크의 오류 발생 가능성 예측하는 것도 가능하다.
- [92]
- [93] 또한, 컨텍스트 서버(58)는 클라이언트 단말(10)과 특정 스크래핑을 위한 세션이 유지되는 동안 동기화를 수행하고 일련의 동작들을 파악하도록 하는 것도 가능하다.
- [94] 세션이 유지되는 동안 컨텍스트 서버(58)는 클라이언트 단말(10)에서 출력되는 태스크 수행 결과값들을 수신하여 로그로 기록할 수 있다. 컨텍스트 서버(58)는 수행 결과들을 로그를 저장함으로써, 스크래핑 과정에서 오류가 발생 시 원인을 쉽게 파악할 수 있도록 한다.
- [95] 종래의 스크래핑 과정에서는 클라이언트 단말(10)에서 오류가 발생 시 그 결과값만 받기 때문에 오류에 대한 세부적인 확인이 어려웠다. 하지만, 컨텍스트 서버(50)는 한 세션 동안의 수행 결과를 로그 형태로 저장함으로써 순차적인 프로세스 중 어떠한 단계에서 에러가 발생하였는지를 쉽게 확인할 수 있도록 한다.
- [96] 또한, 컨텍스트 서버(58)는 로그로 저장되는 정보 중 개인정보가 포함된 경우 일정 보관 기간 후 폐기하도록 규정하여 스크래핑으로 획득되는 개인정보의 유출을 방지하는 것도 가능하다.
- [97]
- [98] 태스크 수행부(110)는 룰 서버(52)로부터 룰 정보를 수신하고, 룰 정보에 따른 태스크를 수행할 수 있다.
- [99] 태스크 수행부(110)는 수신된 룰 정보에 따른 태스크를 수행하되, 태스크 내 정의된 커맨드들 중 일부 커맨드의 경우는 커맨드 서버(54)를 통해 수행되도록 요청할 수 있다.
- [100] 태스크 수행부(110)는 원격 커맨드의 경우 커맨드 서버(54)에 원격 프로시저 콜을 송신하고, 이를 수신한 커맨드 서버(54)의 원격 커맨드 수행부(56)는 원격 커맨드를 수행하고 결과를 다시 태스크 수행부(110)에 리턴 할 수 있다.
- [101] 태스크 수행부(110)는 스크래핑 프로세스를 정의된 룰 정보에 따라 수행하고, 수집된 데이터를 클라이언트 단말(10)을 통해 제공할 수 있다.
- [102]
- [103] 이상의 클라이언트 단말(10)과 원격 서버(50) 간의 통신 과정에서 주고 받는 일련의 데이터들은 압축 후 암호화되어 상호 송수신 될 수 있다.

- [104] 스크래핑 과정에서 획득되는 데이터들은 개인정보를 포함할 수 있으며, 웹사이트(20)의 정보들을 주로 이용하므로 HTML(Hypertext Markup Language) 문서 형태로 구성될 수 있다.
- [105] 따라서, HTML 문서 내에 반복되는 문자열을 위주로 우선 압축하고 암호화하는 방식을 취함으로써 압축효율을 높임과 동시에 암호화되는 데이터의 크기를 줄여 암호화 시간을 줄일 수 있다.
- [106] 나아가, 포함되는 개인정보를 암호화함으로써 개인정보의 유출에 따른 피해를 방지할 수 있다.
- [107]
- [108] 이하, 도 3을 참조하여 클라이언트 단말(10)과 물 서버(52)의 상호 동작에 대하여 보다 상세히 설명한다.
- [109] 도 3은 클라이언트 단말(10)과 물 서버(52) 간의 물 정보를 요청하는 프로세스를 예시하는 도이다.
- [110] 먼저, 클라이언트 단말(10)에 설치된 어플리케이션(1000)은 동작 중에 외부의 웹사이트로부터 데이터의 획득이 필요한 경우 스크래핑을 요청할 수 있다.
- [111] 이러한 경우 클라이언트 단말(10)은 태스크 수행부(110)로 스크래핑을 위한 초기값을 제공한다(32).
- [112] 스크래핑을 위한 초기값은 스크래핑을 위한 식별 정보로 스크래핑 대상 웹사이트의 URL 정보 및 데이터의 획득에 필요한 인증 정보 등 스크래핑에 필요한 각종 정보를 포함할 수 있다.
- [113] 또한, 획득하고자 하는 데이터의 형식이나 종류 등을 정의하는 정보를 포함하는 것도 가능하다.
- [114] 태스크 수행부(110)는 스크래핑 초기값을 이용하여 물 서버에 스크래핑을 위한 스크래핑 물을 요청한다(34).
- [115] 물 서버(52)는 수신된 초기값에 따라 저장된 물 중에서 매칭되는 스크래핑 물을 물 정보로 송신한다(36).
- [116] 물 정보를 수신한 태스크 수행부(110)는, 물 정보 내에 정의된 일련의 태스크를 수행한다. 태스크 수행부(110)는 물 정보 내에 가장 우선하는 태스크를 선정하고 이를 수행할 수 있다(38).
- [117] 이때, 각각의 태스크는 스크래핑을 위한 세부 목적에 따라 복수의 커맨드로 구성된 블록으로 수행될 수 있으며 태스크 수행부(110)는 태스크의 매 수행이 완료 될 때 마다 수행 결과를 평가할 수 있다(40).
- [118] 태스크 수행부(110)는 수행 결과에 이상이 없는 경우 물 정보 내의 다음 태스크를 수행한다.
- [119] 이상의 과정을 반복하여 수행하고 물 정보 내에 정의된 태스크를 모두 수행하고, 획득된 데이터를 스크래핑 결과로 클라이언트 단말에 제공할 수 있다(44).
- [120] 이하, 도 4를 참조하여 태스크 수행부(110)의 구체적인 태스크 수행 과정을

설명한다. 본 실시예에서 태스크는 복수의 커맨드로 구성될 수 있다.

- [121] 이때 커맨드는 클라이언트 단말에서 직접 수행이 가능한 로컬 커맨드와 외부의 원격 서버를 통해 수행하여야 하는 원격 커맨드로 구분될 수 있다.
- [122] 따라서, 태스크 수행부(110)는 태스크 수행 요청을 수신하고(62), 우선적으로 수행하여야 할 커맨드를 선별하여 수행할 수 있다.
- [123] 수행하여야 할 커맨드가 로컬 커맨드인 경우 로컬 커맨드는 클라이언트 단말 상에서 수행될 수 있다(64).
- [124] 특정 커맨드의 수행이 완료되면, 다음에 수행 되어야 할 커맨드가 로컬 커맨드 인지 또는 원격 커맨드 인지를 판단할 수 있다.
- [125] 만약 다음 커맨드가 원격 커맨드인 경우(66)에는 태스크 수행부(110)는 원격 프로시저 콜을 커맨드 서버에 송신할 수 있다(68).
- [126] 원격 프로시저 콜에는 원격 수행에 필요한 커맨드의 내용을 정의하는 정보가 포함될 수 있으며, 원격 프로시저 콜을 통해 커맨드 서버(54)는 원격 커맨드를 수행할 수 있다.
- [127] 커맨드 서버(54)는 원격 커맨드의 수행 후 수행 결과를 클라이언트 단말에 리턴한다(72).
- [128] 이상의 과정은 태스크의 커맨드가 모두 수행될 때까지 반복될 수 있으며, 커맨드의 수행이 완료되면 태스크 수행 결과를 클라이언트 단말에 제공할 수 있다(74).
- [129] 본 실시예에서 태스크는 로컬 커맨드와 원격 커맨드의 조합으로 구성될 수 있으며 클라이언트 단말(10)은 커맨드의 성격에 따라 커맨드 서버(54)에 원격 프로시저 콜을 송신하고 결과를 리턴 받는 과정을 반복하여 수행할 수 있다.
- [130]
- [131] 이하, 도 5를 참조하여 본 실시예에 따른 스크래핑 방법의 프로세스를 보다 상세히 설명한다.
- [132] 도 5는 본 발명의 일 실시예에 따른 클라이언트 단말 상에서 어플리케이션이 스크래핑 서비스를 수행하는 방법을 나타내는 흐름도이다.
- [133] 먼저, 외부 웹 사이트에 대한 스크래핑이 필요한 경우 스크래핑 룰 정보를 요청할 수 있다(S10).
- [134] 어플리케이션(1000)은 클라이언트 단말(10)의 통신 모듈을 통해 스크래핑 초기값을 룰 서버(52)로 전송하는 것을 통해 룰 정보를 요청할 수 있다.
- [135] 룰 정보가 요청되면 룰 서버(52)는 매칭되는 스크래핑 프로세스를 정의하는 룰 정보를 검색하고 룰 정보를 클라이언트 단말(10)로 송신할 수 있다.
- [136] 이때 룰 정보는 미리 결정된 리스트 중에 선택될 수 있으며 또는 매칭되는 룰 정보가 존재하지 않는 경우 룰 서버(52)는 기본 커맨드들의 조합을 통해 태스크를 구성하고 이를 룰 정보로 생성하는 것도 가능하다.
- [137] 또한, 수행 주체에 따라 클라이언트 단말(10)에서 수행가능한 로컬 커맨드와 커맨드 서버(54)에서 수행 가능한 원격 커맨드를 구분하여 룰 정보를 생성하고

이를 제공하는 것도 가능하다.

- [138] 나아가, 룰 정보는 기본 커맨드들의 조합으로 미리 구성되는 것 외에 권한 있는 관리자의 외부 접속을 통해 수정되거나 신규로 생성될 수 있다.
- [139] 즉, 스크래핑 대상인 특정 웹 사이트의 변경이 있는 경우 이에 대응하여 수정될 수 있다.
- [140] 따라서, 어플리케이션 내에 스크래핑 룰이 라이브러리로 규정되는 하드 코딩 방식에 비해 능동적으로 관리자가 스크래핑 대상에 대응할 수 있으며 어플리케이션의 업데이트 없이 사용자는 기존의 서비스들을 그대로 이용할 수 있다.
- [141] 클라이언트 단말(10)은 룰 서버(52)로부터 스크래핑 룰 정보를 수신한다(S20).
- [142] 클라이언트 단말(10) 내 태스크 수행부(110)는 수신된 룰 정보에 정의된 복수의 태스크를 상술한 바와 같이 수행한다(S30).
- [143] 태스크는 순차적으로 수행될 수 있으며, 태스크들은 서로 연관관계를 가지고 이전에 수행된 태스크의 결과를 참조하여 수행될 수 있다.
- [144] 룰 정보 내에 정의된 태스크 명과 관련 변수를 이용하여 이전 태스크의 수행 결과를 참조하여 현재 수행되는 태스크는 결과를 생성할 수 있다.
- [145] 태스크는 복수의 커맨드로 구성되며, 커맨드의 수행 과정을 통해 생성된 태스크 결과는 태스크 별로 저장될 수 있다.
- [146] 또한 커맨드는 로컬 커맨드와 원격 커맨드로 구분되며 각각 클라이언트 단말과 커맨드 서버에서 분산 처리될 수 있다.
- [147] 각각의 커맨드들은 비동기식 폴링(polling) 방식으로 처리될 수 있다.
- [148] 따라서 클라이언트 단말(10)이 미리 정해진 프로토콜 내에서 커맨드를 수행하되, 커맨드가 원격 커맨드인 경우 원격 프로시저 콜에 의해 수행을 요청할 수 있다. 커맨드 서버(54)는 원격 프로시저 콜을 수신하면 원격 커맨드를 수행할 수 있다.
- [149] 또한 본 실시예에 따른 스크래핑 방법은 비동기식 방법으로 태스크 내에 구성된 원격 커맨드의 횟수에 따라 태스크의 수행에 지연이 발생될 수 있다.
- [150] 원격 커맨드는 원격 프로시저 콜의 횟수와 결과를 리턴 받는 횟수를 증가 시킴으로 통신 상태에 따라 태스크의 수행이 지연될 수 있으며, 원격 커맨드의 결과가 수신될 때까지 다른 로컬 커맨드의 수행을 대기시킬 필요가 있을 수 있다.
- [151] 따라서 본 실시예에 따른 스크래핑 방법은 태스크 내의 커맨드 구성을 고려하여 태스크 수행부(110)의 제어권을 일시적으로 커맨드 서버(54)에 전달하는 방식을 이용할 수 있다.
- [152] 이와 관련하여 도 6을 참조하면, 태스크 수행부(110)는 수행되어야 할 커맨드가 로컬 또는 원격 수행 여부를 판단함과 동시에, 원격 커맨드가 연속되는 지 여부를 추가적으로 판단할 수 있다.
- [153] 태스크 수행부(110)는 로컬 커맨드를 수행(82)하되 태스크의 수행 스케줄에

따라 원격 커맨드가 연속되는 경우 커맨드의 수행 제어권을 커맨드 서버(54)에 전달할 수 있다.

- [154] 따라서 제어권을 획득한 커맨드 서버(54)는 원격 커맨드의 수행 마다 결과를 클라이언트 단말(10)에 리턴하고, 새로운 원격 프로시저 콜을 수신하는 반복적인 과정을 생략할 수 있다.
- [155] 즉, 연속적인 원격 커맨드를 별도의 원격 프로시저 콜의 수신 없이 수행되도록 할 수 있다(88, 90). 따라서, 원격 프로시저 콜의 송신과 결과 리턴까지의 대기 시간을 줄일 수 있다.
- [156] 이상의 과정을 통해 연속적인 원격 커맨드를 수행한 경우 커맨드 서버는 수행 결과의 리턴과 함께 제어권을 다시 태스크 수행부에 전달할 수 있다(92).
- [157]
- [158] 또한, 본 실시예에 따른 스케줄링 방법은 태스크 단위로 스케줄링을 수행하되 태스크들의 변수 참조 관계를 고려하여 병렬적으로 태스크를 수행하도록 하여 스케줄링 효율성을 높일 수 있다.
- [159] 상술한 바와 같이 태스크 내 커맨드들의 참조 여부로 태스크 간의 상호 의존 관계를 파악할 수 있으며, 이때의 의존 관계를 이용하여 보다 빠르고 효율적인 스케줄링이 가능하도록 하는 것도 가능하다.
- [160] 태스크 수행부(110)는 전체 태스크의 수행 과정을 고려하여 태스크 간의 입력 변수와 출력 변수가 서로 연관되지 않는 독립된 태스크들은 룰 정보에 따른 순서와 무관하게 병렬 적으로 동시에 수행되도록 할 수 있다.
- [161]
- [162] 예를 들어 룰 정보 내 정의된 태스크들의 의존 관계는 도 7a와 같은 의존성 그래프로 표현될 수 있다. 이때 그래프의 각 노드는 태스크를 의미하고, 노드 간을 연결하는 링크 및 링크의 화살표는 태스크의 참조 방향을 정의할 수 있다.
- [163] 도 7a의 경우 B, C 및 D 태스크는 A 태스크의 수행 결과를 참조하며, E 태스크는 B, C 태스크의 수행 결과를 참조할 수 있다. F 태스크는 E 태스크와 D 태스크의 수행 결과를 동시에 참조한다.
- [164] 따라서, 태스크 수행부(110)는 실제 룰 정보를 생성할 시에는 A, B, C, D, E, 및 F 태스크 순으로 순차적으로 수행되도록 관리자가 코드를 작성하였더라도 실제 태스크의 수행 시에는 최적화된 동작을 위해 의존 관계를 고려하여 수행 순서를 결정할 수 있다.
- [165] 구체적으로 의존 관계에 따라 B, C 및 D 태스크는 A 태스크의 결과를 독립적으로 참조하며, 서로에 대한 참조가 없으므로 동시에 수행하도록 하여, 보다 빠르고 효율적인 스케줄링 작업이 가능하도록 할 수 있다. 즉, 태스크 수행부(110)는 B, C 및 D 태스크가 룰 정보 내에 순서대로 정의되었더라도 실제 수행 시 태스크들을 동시에 병렬로 수행할 수 있다.
- [166] 또한, 룰 서버(52)가 태스크 간의 의존 관계를 미리 파악하고 실제 태스크의 수행 순서를 결정하여 클라이언트 단말(10)에 제공하는 것도 가능하다.

- [167] 따라서, 룰 서버(52)는 도 7a와 같이 정의되는 태스크 들의 의존 관계 통해 룰 정보 태스크의 수행 순서를 재 정의하는 시퀀스 그래프를 도 7b와 같은 형태로 추출할 수 있다.
- [168] 도 7b를 참조하면 시퀀스 그래프는 수평 방향의 링크로 순차 수행이 필요한 태스크들의 수행 순서를 정의하고, 수직 방향의 링크로 동시에 병렬 수행이 가능한 태스크들의 관계를 정의할 수 있다.
- [169] 따라서, 룰 서버(52)는 태스크의 수행 순서를 시퀀스 그래프와 같은 형태로 재 정의하여 클라이언트 단말(10)에 제공함으로써, A 태스크 수행 뒤 B, C, 및 D 태스크 들은 동시에 수행되도록 하여 전체 스크래핑의 효율성을 높일 수 있다.
- [170] 이상의 스크래핑 태스크의 수행이 종료되면 수행 결과로 어플리케이션의 외부 웹 사이트로부터 데이터를 수집한다(S40).
- [171] 수집된 데이터는 어플리케이션(1000)을 통해 사용자가 요청한 서비스에 맞추어 제공될 수 있다(S50).
- [172] 이하, 도 8을 참조하여 본 발명의 일 실시예에 따른 어플리케이션 내 라이브러리의 구성에 대하여 설명한다.
- [173] 일 실시예로서, 어플리케이션은 클라이언트 단말상에 설치되어 구동되는 어플리케이션 프로그램(application program 또는 application software)으로서, 룰 서버 및 커맨드 서버와 연계되어 동작하도록 구현된다.
- [174] 어플리케이션내 라이브러리(100)는 스크래핑 룰 요청부(120), 태스크 수행부(110), 서비스 제공부(130)로 구성될 수 있다.
- [175] 스크래핑 룰 요청부(120)는 룰 서버(52)로 스크래핑을 위한 룰 정보를 요청할 수 있다.
- [176] 즉, 본 실시예에서 어플리케이션은 스크래핑을 위한 프로세스를 대상 웹 사이트 마다 내부에 미리 저장하는 방식(Hard-coded)이 아닌 프로세스를 별도의 룰 서버(52)로 외부에 저장하고 관리하는 방식으로 구현된다.
- [177] 따라서, 스크래핑 대상의 변경 사항에 능동적으로 대응할 수 있으며 별도의 어플리케이션을 재 배포 할 필요가 없이 룰 서버 내 룰 정보의 변경 만으로 대응할 수 있는 장점이 있다.
- [178] 태스크 수행부(110)는 스크래핑 룰 요청부(120)에서 수신한 룰 정보를 이용하여 태스크를 수행할 수 있다.
- [179] 구체적으로 태스크는 커맨드들의 조합으로 구성될 수 있으며, 태스크는 로컬 커맨드와 원격 커맨드로 구성될 수 있다.
- [180] 로컬 커맨드는 클라이언트 단말에서 직접 수행되는 명령으로 예를 들어 기본적으로 발생하는 명령으로 룰 서버 내에서 미리 생성될 수 있다.
- [181] 예를 들어 HTTP 요청/응답, 사용자 입력을 요청하고, HTTP 로그인 요청/응답 등을 수행하는 명령을 로컬 커맨드로 생성될 수 있다.
- [182] 원격 커맨드는 커맨드 서버를 통해 수행되는 명령으로 요청되는 외부 데이터에 따라 맞춤형(Custom)으로 생성될 수 있다.

- [183] 예를 들어 맞춤형 원격 커맨드를 통해 커맨드 서버가 공인 인증서에 대응하여 해시 키를 생성하고, 커스텀 페이지 기반의 정보 페이지를 요청하는 등 부가적인 처리가 필요한 업무를 수행할 수 있다.
- [184] 대상 웹 사이트내 페이지의 갱신이 발생하는 경우 커스텀 페이지로 구성하여 이에 대응한 커맨드를 원격에서 수행하도록 할 수 있다.
- [185] 태스크 수행부(110)는 원격 커맨드의 수행을 위해 원격 프로시저 콜을 요청하는 RPC 송신부(112)와, 원격 커맨드의 수행 결과를 수신하는 커맨드 결과 수신부(114)를 더욱 포함할 수 있다.
- [186] 서비스 제공부(130)는 이상의 태스크 수행부(110)에서 수집된 데이터를 가공하여 사용자에게 제공해 준다.
- [187]
- [188] 이상 본 발명에 따르면, 스크래핑 서비스를 클라이언트 단말과 원격 서버를 통해 분산 제공함으로써 스크래핑 업무의 효율을 높이고, 스크래핑 대상의 변경 사항에 능동적으로 대응할 수 있다.
- [189] 또한, 대상의 변경에 대하여 어플리케이션의 업데이트나 재배포를 최소화하여 사용자의 이용 편의성을 높일 수 있다.
- [190] 또한, 스크래핑을 수행하는 원격 서버를 클라이언트 단말과 독립적으로 운영하여 보안성을 높이고 스크래핑 수행을 위해 이용되는 정보에 대한 외부의 접근과 유출을 방지할 수 있다.
- [191] 이상, 여기에 설명되는 다양한 실시예는 예를 들어, 소프트웨어, 하드웨어 또는 이들의 조합된 것을 이용하여 컴퓨터 또는 이와 유사한 장치로 읽을 수 있는 기록매체 내에서 구현될 수 있다.
- [192]
- [193] *하드웨어적인 구현에 의하면, 여기에 설명되는 실시예는 ASICs (application specific integrated circuits), DSPs (digital signal processors), DSPDs (digital signal processing devices), PLDs (programmable logic devices), FPGAs (field programmable gate arrays, 프로세서(processors), 제어기(controllers), 마이크로 컨트롤러(micro-controllers), 마이크로 프로세서(microprocessors), 기타 기능 수행을 위한 전기적인 유닛 중 적어도 하나를 이용하여 구현될 수 있다. 일부의 경우에 본 명세서에서 설명되는 실시예들이 제어 모듈 자체로 구현될 수 있다.
- [194] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능과 같은 실시예들은 별도의 소프트웨어 모듈들로 구현될 수 있다. 상기 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리 모듈에 저장되고, 제어모듈에 의해 실행될 수 있다.
- [195] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본

발명의 본질적인 특성에서 벗어나지 않는 범위 내에서 다양한 수정, 변경 및 치환이 가능할 것이다.

- [196] 따라서, 본 발명에 개시된 실시 예 및 첨부된 도면들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예 및 첨부된 도면에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구 범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리 범위에 포함되는 것으로 해석되어야 할 것이다.