Model-Driven Performance Analysis of Large Scale Irrigation Networks

Muhammad Umer Tariq†, Hasan Arshad Nasir§
†‡*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA.*
†*m.umer.tariq@gatech.edu*, §*hasan.nasir@lums.edu.pk*

Abubakr Muhammad♯, Marilyn Wolf‡
§♯*Department of Electrical Engineering, LUMS, School of Science and Engineering, Lahore, Pakistan.*
♯*abubakr@lums.edu.pk*, ‡*wolf@ece.gatech.edu*

***Abstract*—Irrigation networks play a fundamental part in the agriculture system of various countries. In the wake of global environmental challenges and economic competition, efficient use of water resources has become extremely important. This can only be achieved by developing smarter control infrastructures for irrigation networks, via the incorporation of communication and computation technologies. Thus, future irrigation networks represent a prime example of cyber physical systems. Effective operation of these complex cyber physical systems is not possible with conventional methods and requires unprecedented levels of automation and decision-support tools. We argue that these cyber physical systems will require a complete model-driven toolset for effective operation. As a first step towards that tool flow, we have developed a model-driven simulation infrastructure for irrigation networks. In the future, we propose to complete the toolset by developing a model-driven configuration infrastructure. Our contributions in this paper include the development of a domain-specific modeling language (DSML) for irrigation networks, implementation of this DSML in Generic Modeling Environment (GME), and automatic simulator M-file generation capability from the DSML-based case diagram of an arbitrary irrigation network. Moreover, we present case studies of water distribution and flood management to show the utility as well as the effectiveness of our approach. We also present the performance of our toolset for the realistic scenario of irrigation networks in Pakistan.**

***Keywords*-Irrigation Networks; Cyber Physical Systems; Modeling and Simulation; Model-Driven Software Development; Domain Specific Modeling Languages; Model Transformation;**

I. Introduction

In the wake of continuously increasing global population, food security is one of the major challenges facing humanity today. Irrigation networks play a pivotal role in the agriculture system of numerous countries. Increased competition in the global marketplace and recent awareness about global environmental challenges has resulted in a demand for more efficient handling of these irrigation networks. This can only be achieved by upgrading the control mechanisms for these irrigation networks with the advances made in the field of sensing, communication and computation technologies. There has been a dramatic decrease in the cost of these technologies, making them feasible for large-scale applications. This potential combination of communication, computation, and control makes the future irrigation networks a prime example of cyber physical systems. Cyber physical systems (CPS) are a new generation of engineering paradigms, capable of deploying, monitoring, and controlling large-scale infrastructures (both natural and man-made) by a tight integration of computational and physical processes [23] [28]. The hallmark of CPS is a pervasiveness of networked embedded devices – sensing and autonomously manipulating large-scale physical processes in a distributed manner [20] [11].

We envision that these future irrigation networks will be under the control of mechanized gates, operated remotely from a control center. Safe and reliable operation of these irrigation networks requires that their development be accompanied by a sound set of tools for their operation. One potential paradigm for the development of such tools is Model-Driven Development (MDD). MDD [24] is a design paradigm in which high-level models are transformed into low-level models through the process of model transformation. MDD has been applied in other domains such as automotive development [3] and power system analysis [6]. However, in these state-of-the-art systems, use of model-driven paradigm is only limited to development and analysis phases. We propose to develop an integrated model-driven approach that encompasses design, development, analysism and operation phases of a cyber physical system. In this research paper, we present the steps taken towards the goal of a comprehensive model-driven approach to CPS, in the context of automated irrigation networks.

In this paper, we argue that a model-driven tool set, that provides: a) a domain-specific modeling language for developing the case diagram of actual irrigation networks, b) a model-driven simulation, and c) model-driven configuration capability, can be an invaluable asset for the effective and reliable operation of the future irrigation networks [9] [24]. In this paper, we present such a model-driven toolset, with a DSML and a model-driven simulation capability for irrigation networks. We present the utility of this model-driven performance analysis tool in the case of water distribution under normal circumstances and flood management. Our contributions in this paper include:

1) The development of a DSML [24] for irrigation networks in GME [10] that allows the user to develop

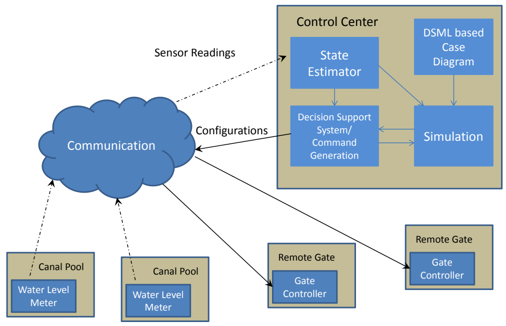

Figure 1. Envisioned irrigation network control infrastructure.

case diagrams for an arbitrary irrigation network

2) A simulation engine for a complete irrigation network with certain simplifying assumptions that is an extension of work presented in [16].
3) A Builder Object Network (BON) [8] component interface-based generative capability that can generate M-files for the simulation engine, from the case-diagrams of arbitrary irrigation networks, thus allowing complete automation of their simulation and subsequent performance analysis during operations management.

However, the paper does not claim to provide a complete solution to the problem of automation of future irrigation networks. We intend to extend this work with a capability of generating configurations for remote mechanized gate controllers from the case-diagrams in our toolset presented in this paper. Fig. 1 illustrates the envisioned automated control infrastructure for the future of irrigation networks.

The rest of the paper is organized as follows. Section II presents some related work. Section III presents the details of the DSML developed for the irrigation networks. Section IV presents the implementation of this DSML in GME. Section V describes the development of M-file generation capability from a case diagram in GME, through the BON component interface of GME. Section VI presents the details of the simulation engine for large irrigation networks. Section VII presents the application of our toolset to the problems of water distribution and flood management. Section VIII presents the conclusion and future work.

II. Related Work

GME and its model-interpretation capability has been used in various other domains such as networked control systems [21] and reconfigurable conveyer systems [2]. However, its use is usually limited to off-line analysis or off-line synthesis, through its integration with different analysis and synthesis tools. However, in our work, we aim to use the capabilities of GME in a broader context by designing our DSML and model interpreters in such a way that our toolset not only supports the analysis of irrigation network but also its configuration through the generation of control commands for various remotely located gates.

A related problem of performance of metropolitan water distribution network has been tackled in [12]. This work uses EPANET [22] and MATLAB as the underlying simulation engine. However, it does not use a systematic model-based approach for such analysis that can be really useful in order to take this work beyond offline analysis and into the development of decision-support tools for operations management staff, handling this infrastructure.

There exists a vast body of literature on modeling the flow of water in open channels [5] [26] and controlling water levels in an irrigation canal network using methods of feedback control [13] [4] [19]. Various control-theoretic problems have been addressed and solved by researchers related to modeling and controller design [18], decentralization [7] [4] [17], security [1], and practical implementation issues [14] [18] [25]. In our previous work [16], we have proposed that the problem of efficient distribution of available water can be effectively solved by engineering appropriate cyber physical systems (CPS) and feedback control technologies.

III. Domain-Specific Modeling Language (DSML)

In this section, we present the building blocks of our DSML, developed for representing large-scale irrigation networks as case-diagrams in GME. Main building blocks are *PrimaryCanalPool*, *SecondaryCanalPool*, *TertiaryWaterChannel*, *Gate*, *Controller*, and *LevelMeter*. There are two types of connections: *PhysicalLinks* and *CommunicationLinks*. Fig. 2 shows a snapshot of a typical irrigation network case-diagram in our GME-based toolset. Each building block allows us to set a number of attributes. These attributes are later used in the simulation which is set up through the M-file generative capability of our toolset.

A. Primary Canal Pool

Primary Canal is a canal that is taken directly out of the river. It is also sometimes referred to as Main Canal. This canal is divided into multiple pools by gates that control the flow of water from one pool to another. We use *PrimaryCanalPool* as a building block in our DSML. Some of the important attributes of the *PrimaryCanalPool* are: *LengthOfPrmaryCanalPool*, *WidthOfPrimaryCanalPool*, *BedSlopeOfPrimaryCanalPool*, *InitialWaterLevel*, *InitialWaterFlow*, *SetPoint*, *PrimaryCanalPoolAddress*, and *LocationOfWaterLevelMeter*.[1]

[1]Throughout this paper we have assumed channels of rectangular cross section. However this is not a limitation both in representation or simulation.

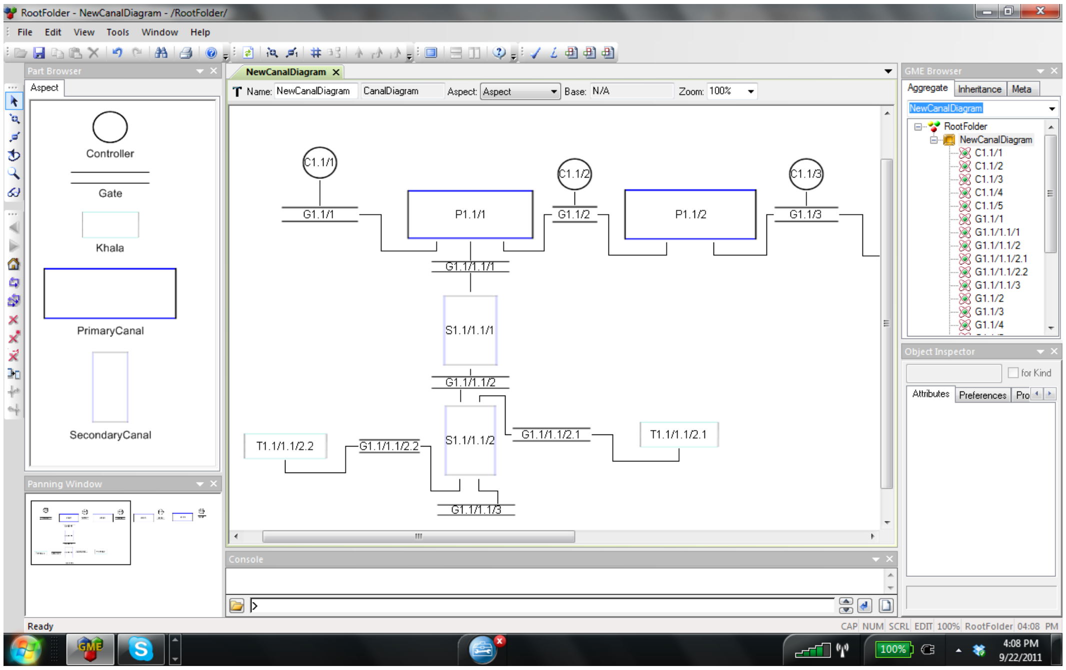

Figure 2. A typical irrigation network model in our DSML mentioned prosumer interactions.

B. Secondary Canal Pool

Secondary Canal is a canal that is taken out of one of the Primary Canal pools. It is also sometimes referred to as Distributory Canal. More than one secondary canals can be taken out of a single primary canal pool. Secondary Canal can also be divided into multiple pools by gates that control the flow of water from one pool to another. We use *SecondaryCanalPool* as a building block in our DSML. Some of the important attributes of the *SecondaryCanalPool* are: *LengthOfSecondaryCanalPool*, *WidthOfSecondaryCanalPool*, *BedSlopeOfSecondaryCanalPool*, *InitialWaterLevel*, *InitialWaterFlow*, *SetPoint*, *SecondaryCanalPoolAddress*, and *LocationOfWaterLevelMeter*.

C. Tertiary Water Channel

Tertiary Water Channel (also commonly known as "Khala" in colloquial Punjabi) is taken out of a secondary canal through a gate which controls the flow of water in it. Usually, tertiary water channels are ungated and thus not divided into pools. We use *TertiaryWaterChannel* as a building block in our DSML. Some of the important attributes of *TertiaryWaterChannel* are: *LengthOfTertiaryWaterChannel*, *WidthOfTertiaryWaterChannel*, *BedSlopeOfTertiaryWaterChannel*, *InitiaWaterLevel*, *InitialWaterFlow*, *SetPoint*, *TertiaryWaterChannelAddress*, and *LocationOfLevelMeter*.

D. Gate

Gates are used at numerous places throughout the irrigation network. They are used to control the flow of water from one primary canal pool to another, from primary canal to a secondary canal, from one secondary canal pool to another, and from secondary canal to a tertiary water channel. We use *Gate* as a building block in our DSML. Some of the important attributes of *Gate* are: *InitialHeightOfGate*, *FinalHeightOfGate*, *InitialHeadAboveGate*, *FinalHeadAboveGate*, and *GateAddress*.

E. Controller

Every gate in the irrigation network has an associated controller. However, the level of sophistication of the controller can vary from a simple manual controller to a remotely configurable automated controller. We use *Controller* as a building block in our DSML. Some of the important attributes of *Controller* are: *ControllerAddress*, *ControllerParameters*, and *ControllerProfile*. The attribute *ControllerProfile* refers to an XML document that contains information about communication and computation capabilities of the controller. This information will be important when this DSML is used in a tool that automatically generates configurations for a remote gate controller.

F. Level Meter

Water Level Meter is an important sensing component of irrigation network. Its reading is used by the controllers to maintain required water levels at specified points throughout the irrigation network. We use *LevelMeter* as a building block in our DSML.[2]

IV. GME Implementation of DSML

Generic Modeling Environment (GME) [10] is a configurable modeling environment that can be configured to support a certain DSML. Configuration of GME is not optional rather it is the first step before anything meaningful can be done in this environment. This configuration is known as metamodeling [10]. A metamodeling expert needs a specification of the domain-specific modeling language (DSML). GME supports a set of generic modeling concepts that are variations of the UML's entities, relationships, and attributes. These generic modeling concepts (or First Class Objects, FCOs) are Atoms, Models, Connections, References and Sets [10]. During metamodeling, a suitable GME concept is chosen to represent each of the concepts that appears in the specification of DSML. Once a metamodel has been developed, it is used by *metaGMEinterpreter* to register a new modeling paradigm in GME that supports the desired DSML. After this step, GME is ready to develop models in the desired DSML. GME also provides a COM-based C/C++ interface to its models that can be used to develop model interpreters. These model interpreters can read the information embedded in models developed in DSML and generate appropriate output for various analysis tools and execution environments. Fig. 3 shows a portion of the metamodel developed in GME for our DSML described in Section III. During the metamodeling, certain building blocks of DSML such as *PrimaryCanalPool*, *SecondaryCanalPool*, and *TertiaryWaterChannel* were treated as Models since they are supposed to have another building block inside i.e. *LevelMeter*. On the other hand, *LevelMeter*, *Gate*, and *Controller* were treated as Atoms in the metamodel since they do not have further elements inside them. Moreover, the metamodel shows that connection *PhysicalLink*, always has the Gate as its source and the destination could be any of the *PrimaryCanalPool*, *SecondaryCanalPool*, *TertiaryWaterChannel*, or *Controller*. The connection *CommunicationLink* is always between a *LevelMeter* and a *Controller*.

V. M-File Generation

In section IV, we described how GME can be configured for modeling in a DSML through the process of metamodeling. Besides providing this capability, GME also provides a Builder Object Network (BON) component interface that allows a C/C++ program to access the information about models developed in GME. This capability can be used for developing model interpreters in C/C++ development environment. C/C++ programs can read the information embedded in GME models through the BON component interface methods (such as *GetRootModels()*, *GetChildren()*, *GetModels()*, *GetAtoms()*, *GetConnections()*, *GetKindName()*, *GetAttribute()*), and generate appropriate output for various analysis tools and execution environments. Subsequently, these C/C++ programs can be compiled and registered with GME as model interpreters. Using the model-interpreter capability of GME and its BON component interface, we have developed a generative capability in our toolset. This allows us to take an arbitrary case-diagram of an irrigation network and generate the M-file for the simulation of that particular scenario according to our underlying irrigation network simulation engine (described later). This automation frees the operator from the infeasible task of writing an M-file for each of the different scenarios faced during the operation of a nationwide irrigation network. Fig. 4 shows the overall structure of our GME/MATLAB-based model-driven performance analysis tool.

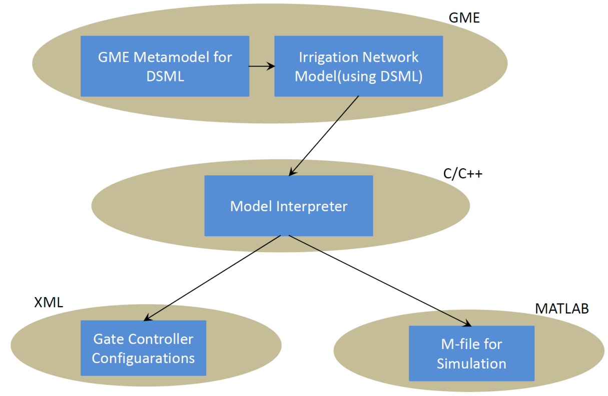

Figure 4. Architecture of proposed model-based toolset

VI. Irrigation Network Simulation Engine

The underlying simulation engine comprises rigorous modeling of unsteady flow of water in open channels and numerical solution to that model. This model is based on Saint Venant's equation (SVE) which are derived from a mass and momentum balance, and are given by:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0,$$

$$\frac{\partial Q}{\partial t} + (\frac{gA}{B} - \frac{Q^2}{A^2})\frac{\partial A}{\partial x} + \frac{2Q}{A}\frac{\partial Q}{\partial x} + gA(S_f - S_0) = 0. \quad (1)$$

Here x is the distance coordinate; t is time; A is the cross-sectional area of the channel; B is the width of water surface; Q is the flow (discharge); g is the gravitational constant (taken as $9.81m/s^2$); S_f and S_0 are frictional and bed slope of the channel respectively. For a section of a channel, these variables have been described in Fig. 5.

[2] In many developed countries flow meters and other type of sensors are used. However, in countries like Pakistan, only level meters are mostly installed. That is why, we have not used other sensors, but this is not a limitation.

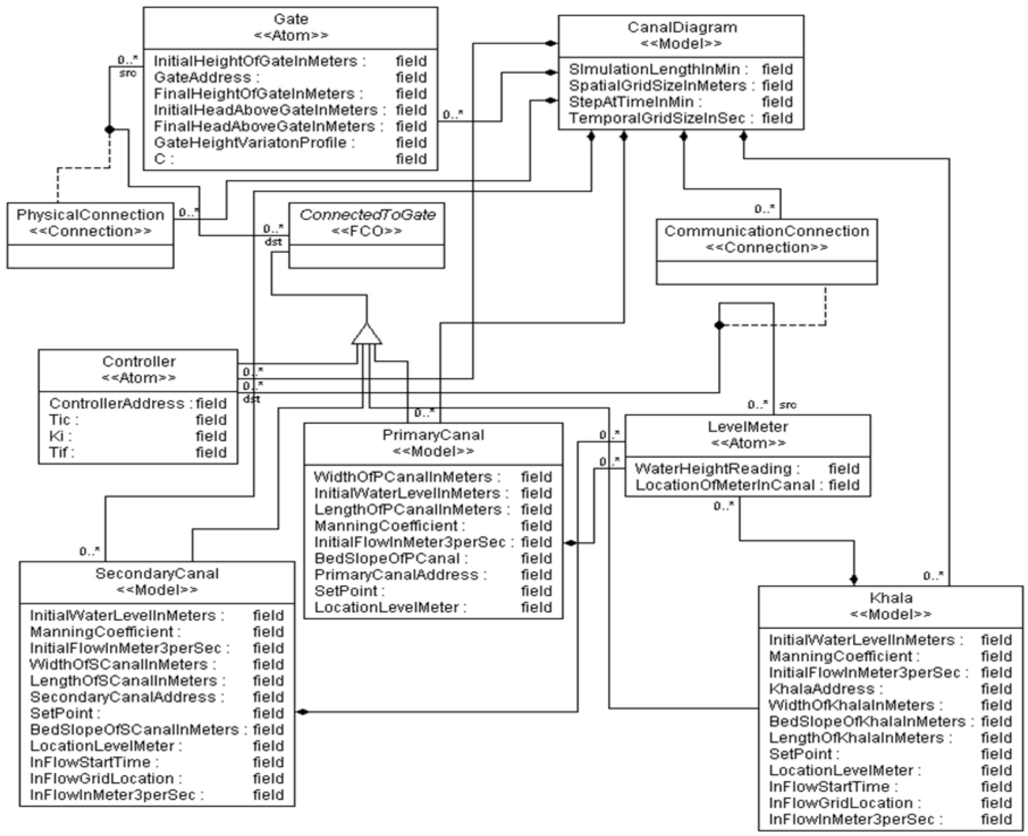

Figure 3. GME metamodel for the proposed DSML.

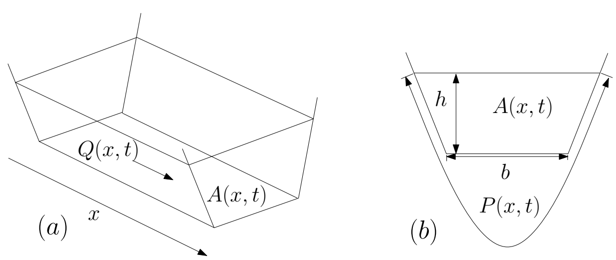

Figure 5. (a) open channel flow along a longitudinal axis indexed by abscissa x. (b) cross-section of an open trapezoidal channel

A. Numerical Solution of St. Venant PDEs

The *Preissmann's scheme* is commonly accepted to be the most robust finite-difference scheme for solving the system of SVEs [26]. The time, t, and spatial variable, x, are discretized in the Preismann's scheme onto a grid on which the SVEs are approximated using approximations of partial derivatives presented in [15]. Let us assume a rectangular channel of bottom width B. At any time t, and distance, x, the height of water is given by $h(x,t)$ and the cross sectional area of water channel is $A(x,t) = Bh(x,t)$. Also using Preissmann's scheme approximation and Manning formula [5] to Eq. 1, we get Eqs. 2 and 3. The system is solved for $0 \leq x \leq L$ and $t \geq 0$. Initial and final boundary conditions are given by:

- $h(x,t=0)$ and $Q(x,t=0)$ for $0 \leq x \leq L$.
- $Q(x=0,t) = Q_0(t)$ and $Q(x=L,t) = Q_L(t)$ for $t \geq 0$.

$$\begin{aligned}&\tfrac{1}{2}\left(\frac{h_i^{k+1}-h_i^k}{\Delta t}+\frac{h_{i+1}^{k+1}-h_{i+1}^k}{\Delta t}\right)+\\&\tfrac{1}{B}\left((1-\alpha)\left(\frac{Q_{i+1}^k-Q_i^k}{\Delta x}\right)+\alpha\left(\frac{Q_{i+1}^{k+1}-Q_i^{k+1}}{\Delta x}\right)\right)=0.\end{aligned} \tag{2}$$

$$\begin{aligned}&\tfrac{1}{2}\left(\frac{Q_i^{k+1}-Q_i^k}{\Delta t}+\frac{Q_{i+1}^{k+1}-Q_{i+1}^k}{\Delta t}\right)+gBh_P\left((1-\alpha)\left(\frac{h_{i+1}^k-h_i^k}{\Delta x}\right)\right.\\&\left.+\alpha\left(\frac{h_{i+1}^{k+1}-h_i^{k+1}}{\Delta x}\right)\right)+(1-\alpha)\left(\frac{(\frac{Q^2}{Bh})_{i+1}^k-(\frac{Q^2}{Bh})_i^k}{\Delta x}\right)+\\&\alpha\left(\frac{(\frac{Q^2}{Bh})_{i+1}^{k+1}-(\frac{Q^2}{Bh})_i^{k+1}}{\Delta x}\right)+\left(\frac{gn^2Q^2}{BhR^{\frac{4}{3}}}\right)_P=0.\end{aligned} \tag{3}$$

Here, subscript i represents the spatial grid point and superscript k represents the grid point in time. Δx and Δt are grid intervals along x and t axes respectively. In Eqs. 2 and 3, the water levels at node $k+1$ i.e. h_i^{k+1} and h_{i+1}^{k+1} and the flow rates at node $k+1$ i.e. Q_i^{k+1} and Q_{i+1}^{k+1} are unknown. This pair of equations can be written for all spatial grids i.e.

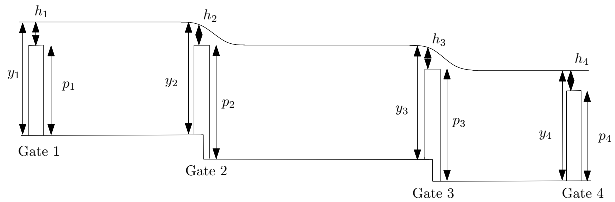

Figure 6. Side view of pools and gates in an open channel.

$i = 1, 2, 3, ..., M-1$. We therefore obtain $2(M-1)$ algebraic equations with $2M$ unknowns. Two boundary conditions of flow rates complete the system as, $Q_1^{k+1} = Q_0(t_{k+1})$. and $Q_M^{k+1} = Q_L(t_{k+1})$. Finally, a system of $2M$-dimensional non-linear algebraic equations in $2M$ variables is obtained. This system of equations is solved using Newton-Raphson method. The detailed solution of these equations can be found from our previous work [16].

B. Channel Geometry and Physical Parameters

Let us consider a (rectangular) channel comprising many interconnected pools, each with bottom slope nearly zero. Each pool has an upstream and downstream gate associated with it to control the amount of water flowing into it from its upstream pool. A three pool system is depicted in Fig. 6. In Fig. 6, y_m, represents height of water in pool m at gate position, h_m, depicts water head over the gate, $p_m = y_m - h_m$, shows height of the m-th gate. For a sharp-edged rectangular gate, flow, Q_m, over any gate can be described by an empirical formula which results in boundary conditions:

$$Q_{in} = c_{in} h^{3/2}, \qquad Q_{out} = c_{out} h^{3/2}. \tag{4}$$

where, c_{in} and c_{out} are constants and depend on gate dimensions and $h = y - p$, is the head above the gate.

C. Gate Controller Design

To develop gate controllers, we use a finite-dimensional lumped-parameter ODE model approximation as discussed in [18] [4]. This model can be written as:

$$\dot{y}_{i+1}(t) = c_{i,in} h_i^{3/2}(t - \tau_i) - c_{i+1,out} h_{i+1}^{3/2}(t) \tag{5}$$

We can design a compensator for the canal model in which a PI controller is augmented with a low pass filter. Such a controller $C_i(s)$ is given by:

$$C_i(s) = (\frac{K_i}{T_{i,c}s})(\frac{1 + T_{i,c}s}{1 + T_{i,f}s}). \tag{6}$$

Parameters K_i, $T_{i,c}s$ and $T_{i,f}s$ are tuned based on the procedure given in [18]. Fig. 7 shows how controller is configured on a single pool in canal. Same procedure can be used to build controllers for cascaded pools.

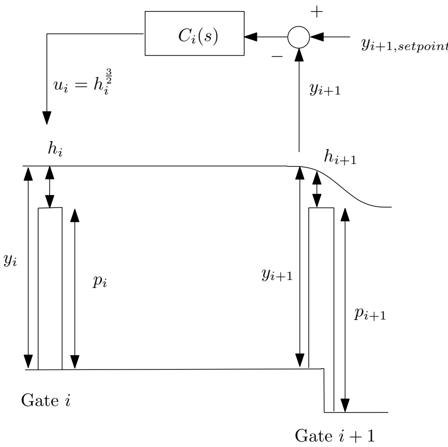

Figure 7. Controller inserted to a single pool

VII. Applications

In this section, we present the usefulness of our model-driven performance analysis tool as a decision-support system during the operation of irrigation networks. We present two case studies: flood level prediction and controlled water distribution. It must be noted, however, that the purpose of this section is to demonstrate potential use cases for our model-driven toolset. We do not concentrate on the performance of simulation engine itself since, in future, we plan to extend our toolset to support multiple commercial off-the-shelf simulation engines.

A. Flood Level Prediction

This model-driven performance analysis toolset can be applied to estimate the flood, its flow and water level in time and over distance. Just to give an example, we have picked a real scenario from irrigation network of Pakistan. Under DG Khan Zone, A Primary Canal emerges from Indus River, named Muzaffargarh Canal [3]. Fig. 8 shows an approximate starting portion of this canal tree structure. Physical parameters of that canal structure are illustrated in Table. I and Table. II [4]. We represented this scenario in our GME-based model-driven toolset using our DSML. Then, we generated the M-File for simulation of this scenario using the model interpreter-based generative capability of our toolset. MATLAB simulations were run for simulation time of 1000 minutes. Grid step sizes, $\Delta x = 100m$ and $\Delta t = 60sec$, were kept in order to preserve the stability of Preissmann's Scheme. The height of all the primary canal pools' gates was $0.8m$. Gates of the first secondary canal pools had a height of $0.6m$, while gates of the other secondary canal, emerging as 'Magasson Branch', had a height of $0.7m$. Height of gate between P1.1 and S1.1.1.1

[3]DG Khan was declared as a calamity-hit area, due to the devastating flood in August 2010.

[4]Parameters are estimated from GIS data

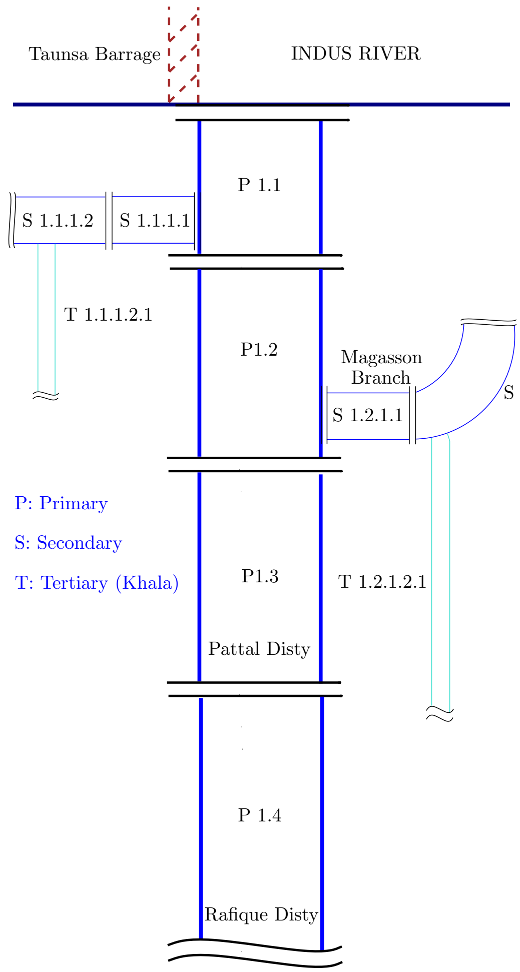

Figure 8. Muzaffargarh Canal starting portion in DG Khan zone

was $0.5m$. Height of gate between P1.2 and S1.2.1.1 was $0.8m$. At time 20 minutes, $0.6m$ head of water (above the gate) was created at the upstream gate of P1.1, and it remained till the end of simulation. This head of $0.6m$, allowed us to simulate a significant flood situation for this portion of the irrigation network. Tertiary water channel (khala), T 1.1.1.2.1, was operated(water was released into it) at time 500 min while the tertiary water channel(khala), T 1.2.1.2.1, was operated at time 800 min. Now, we present the results of our simulation for the scenario, described above.

1) Results: Simulation results make good sense. In Fig. 9, variation of water level is shown in pools P 1.1 and P 1.2. Since head rises at time 20 min., so initially, a fall is observed in hydro graph of pool P 1.1 as water started moving to secondary canal S 1.1.1.1 without any inflow from the river to pool P 1.1. However, after the rise of the head at the upstream gate of pool P1.1 (at time 20 min), water level started rising in pool P1.1, as shown. Pool P 1.2, is fed by its upstream pool i.e. P 1.1, so it experiences increase in water level. Two downstream pools, P1.3 and P1.4, showed very small variations in water level, since water could not travel that much distance in our simulation time of 1000 minutes. Fig. 10 describes variation of water level in secondary pools S 1.1.1.1 and S 1.1.1.2. S 1.1.1.1 was fed by P 1.1 and it continuously experienced increase in water level. However, the flood water could not reach up to 800m distance in S 1.1.1.2, but since tertiary water channel(khala), T 1.1.1.2.1, was operated at time 500 min, so S1.1.1.2 started facing scarcity, as shown in Fig. 10. Fig. 11 tells the story of secondary pools S 1.2.1.1 and S 1.2.1.2. S 1.2.1.1 received water from P 1.2 at time approximately 270 min, after it crossed the gate height of 0.8m. After that, water level in S1.2.2.1 had a continuous rise. At 500m distance in pool S 1.2.1.2, water reached at time, 620 min, but it got a sudden fall in hydrograph, as tertiary water channel(khala), T 1.2.1.2.1, was operated at time, 800 min. Since this simulation is over the region of approximately 25 Km water course and it runs for a simulation time of 1000 minutes, it took some 4 to 5 hours in completing this simulation on a laptop, with common state-of-the-art specifications.

Table I
PHYSICAL DATA OF PRIMARY POOLS IN FIG. 8.

Pool	P 1.1	P 1.2	P 1.3	P 1.4
Length, $L(m)$	1300	4600	6000	7500
Bottom Width, $b_0(m)$	70	50	50	50
Bottom Slope, S_0	0	0	0	0
Gravity, $g(m/s^2)$	9.81	9.81	9.81	9.81
Manning Coefficient, n	0.02	0.02	0.02	0.02
$cin = 0.6\sqrt{g}b_u$	3.6	4.8	4.8	4.8
$cout = 0.6\sqrt{g}b_d$	4.8	4.8	4.8	4.8
Initial Water Level, $y(m)$	0.79	0.79	0.795	0.795
Initial Water Flow, $Q(m^3/s)$	0	0	0	0

Table II
PHYSICAL DATA OF SECONDARY POOLS IN FIG. 8.

Pool	S1.1.1.1	S1.1.1.2	S1.2.1.1	S1.2.1.2
Length, $L(m)$	1000	1000	800	700
Bottom Width, $b_0(m)$	25	25	25	25
Bottom Slope, S_0	0	0	0	0
Gravity, $g(m/s^2)$	9.81	9.81	9.81	9.81
Manning Coefficient, n	0.02	0.02	0.02	0.02
$cin = 0.6\sqrt{g}b_u$	3.6	4	3.6	4
$cout = 0.6\sqrt{g}b_d$	4	4	4	4
Initial Water Level, $y(m)$	0.595	0.595	0.695	0.695
Initial Water Flow, $Q(m^3/s)$	0	0	0	0

B. Controlled Water Distribution

We used our model-driven performance analysis tool for closed loop simulations on same canal as in Fig. 8, except that now we considered only two primary pools and their physical parameters, as mentioned in Table I and Table II. Moreover, Pool P 1.2 was terminated at 1300m. For building gate controllers, we have followed the process mentioned before in Section VI. MATLAB simulations were run for 2000 minutes span. Initial water level in all the pools was

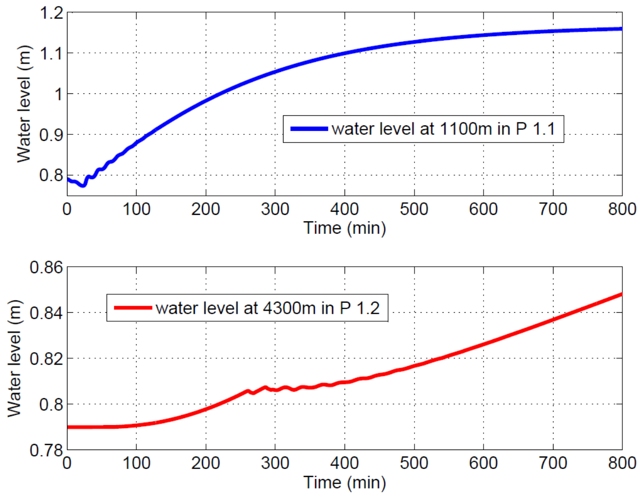

Figure 9. Hydro graph of P 1.1 and P 1.2

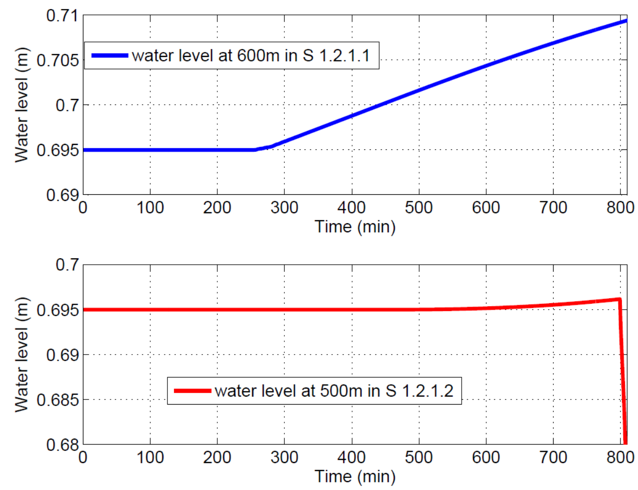

Figure 11. Hydro graph of S 1.2.1.1 and S 1.2.1.2

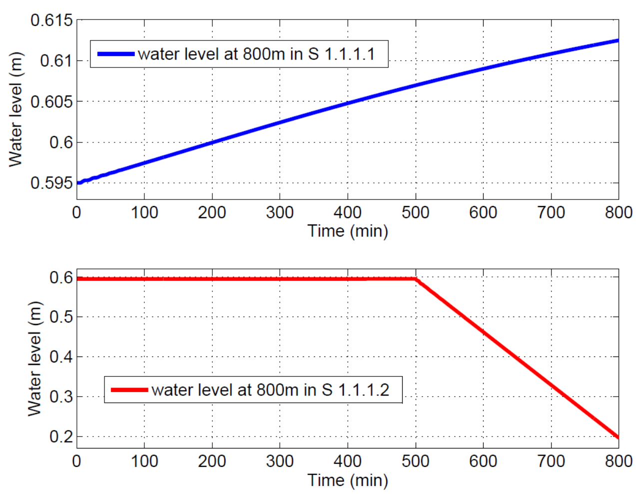

Figure 10. Hydro graph of S 1.1.1.1 and S 1.1.1.2

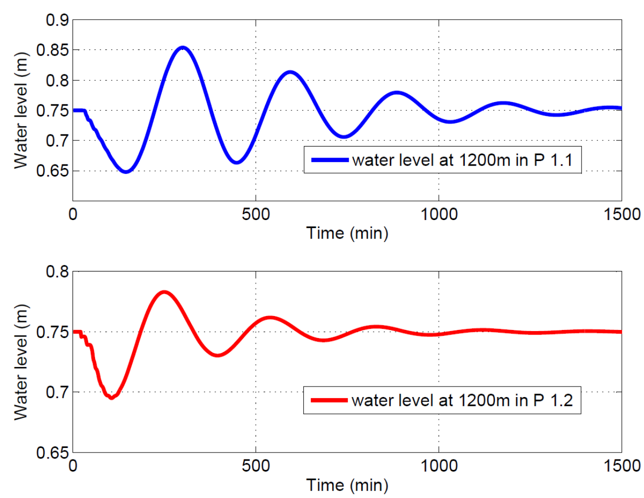

Figure 12. Controlled hydro graph of P 1.1 and P 1.2

kept at 0.75m (set point) and initially all the gates were at height, 0.8m. The aim of our gate controllers was to maintain the water level in pools P1.1, P1.2, S1.1.1.2 and S1.2.1.2, at 0.75m. At time 20 min., downstream gates of pools P 1.2, S 1.1.1.2 and S 1.2.1.2 were opened, this resulted in the development of a scarcity which traveled upstream and after some time, all the pools came back to their steady level of 0.75m through control actions.

1) Results: Fig. 12 and Fig. 13 describe behavior of water level in pools P 1.1, P 1.2, S 1.1.1.2 and S 1.2.1.2. In the figure(s), water level is displaced from set point i.e. 0.75m and then through control actions, this level is restored. The settling time of most upstream pool i.e. P 1.1 is the maximum. This is logically true as it was supposed to feed all downstream pools first and then fulfill its own scarcity.

VIII. Conclusion and Future Work

We have proposed a model-driven toolset for the effective operation of future irrigation networks that will be a prime example of cyber physical systems, by virtue of being controlled by a control infrastructure that incorporates the readily available communication and computation technologies. In this paper, we present a DSML for these irrigation networks and a model-driven performance analysis tool. We present the utility of our toolset by applying it to case studies of flood management and water distribution for representative scenarios from the irrigation network of Pakistan. Our work in this paper is generic enough that our model-driven performance analysis tool can be extended to incorporate other simulation engines than the one used by us.

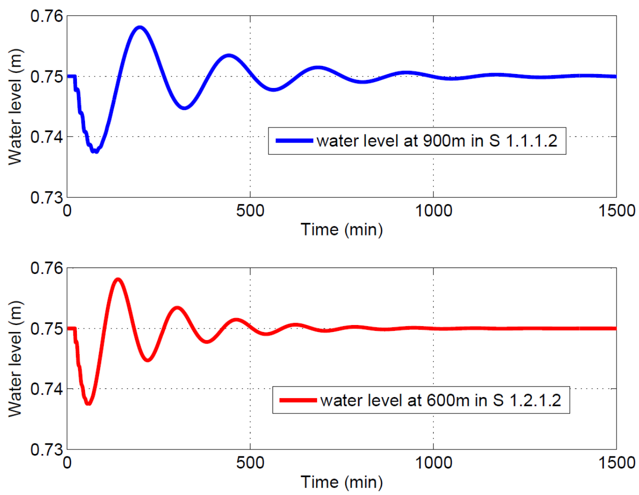

Figure 13. Controlled hydro graph of S 1.1.1.2 and S 1.2.1.2

In future, we plan to extend our current toolset with the model-driven configuration of a distributed set of controllers, controlling the individual mechanized gates. These model-driven performance analysis and model-driven configuration tools, when combined, can form the backbone of the control infrastructure for future irrigation networks, as envisioned, in Fig. 1. For the proposed model-driven configuration capability, we plan to integrate the research presented in this paper with our ongoing work on a time-sensitive service-oriented middleware for a set of distributed controllers [27]. We plan to develop a model transformation from our irrigation netwrok DSML to service-definitions in our service-oriented middleware for a set of controllers. Moreover, in our current implementation of the model-driven toolset, we have used a direct-manipulation approach to model-transformation step that directly transforms our DSML-based irrigation network case diagrams into M-files for our MATLAB-based simulator. However, in future, we plan to define an intermediate format, for describing case diagram information, that will not only simplify the model-transformation process, but will also allow us to integrate other irrigation-network simulators into our model-driven toolset.

References

[1] S. Amin, X. Litrico, S.S. Sastry, and A.M. Bayen. Stealthy deception attacks on water SCADA systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 161–170. ACM, 2010.

[2] K. An. *Model-driven Performance Analysis of Reconfigurable Conveyor Systems used in Material Handling Applications*. PhD thesis, Vanderbilt University, 2011.

[3] E. Bringmann and A. Kramer. Model-based testing of automotive systems. In *Software Testing, Verification, and Validation, 2008 1st International Conference on*, pages 485–493. Ieee, 2008.

[4] M. Cantoni, E. Weyer, Y. Li, S.K. Ooi, I. Mareels, and M. Ryan. Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95(1):75–91, 2007.

[5] M.H. Chaudhry. *Open-channel flow*. Springer Verlag, 2008.

[6] J.D. Glover, M.S. Sarma, T.J. Overbye, and T.J. Overbye. *Power system analysis and design*. Cengage Learning, 2008.

[7] M. Gómez, J. Rodellar, and J.A. Mantecon. Predictive control method for decentralized operation of irrigation canals. *Applied Mathematical Modelling*, 26(11):1039–1056, 2002.

[8] Vanderbilt University Institute for Software Integrated Systems. Generic modeling environment 5 users manual. Version 5.0, 2005.

[9] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty. Model-integrated development of embedded software. *Proceedings of the IEEE*, 91(1):145–164, 2003.

[10] A. Ledeczi, M. Maroti, A. Bakay, G. Karsai, J. Garrett, C. Thomason, G. Nordstrom, J. Sprinkle, and P. Volgyesi. The generic modeling environment. In *Workshop on Intelligent Signal Processing, Budapest, Hungary*, volume 17. Citeseer, 2001.

[11] E.A. Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, 2008.

[12] J. Lin, S. Sedigh, and A. Miller. Towards integrated simulation of cyber-physical systems: a case study on intelligent water distribution. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 690–695. IEEE, 2009.

[13] X. Litrico and V. Fromion. *Modeling and control of hydrosystems*. Springer Verlag, 2009.

[14] X. Litrico, V. Fromion, J.P. Baume, C. Arranja, and M. Rijo. Experimental validation of a methodology to control irrigation canals based on Saint-Venant equations. *Control Engineering Practice*, 13(11):1425–1437, 2005.

[15] D.A. Lyn and P. Goodwin. Stability of a general Preissmann scheme. *Journal of Hydraulic Engineering*, 113(1):16–28, 1987.

[16] H. Nasir and A. Muhammad. Control of very-large scale irrigation networks: A cps approach in a developing world setting. In *World Congress IFAC*, volume 18, pages 10739–10745, 2011.

[17] R.R. Negenborn, P.-J. van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneous Media*, 4(2):359–380, June 2009.

[18] S.K. Ooi and E. Weyer. Control design for an irrigation channel from physical data. *Control Engineering Practice*, 16(9):1132–1150, 2008.

[19] T. Rabbani, S. Munier, D. Dorchies, P. Malaterre, A. Bayen, and X. Litrico. Flatness-based control of open-channel flow in an irrigation canal using SCADA. *Control Systems Magazine, IEEE*, 29(5):22–30, 2009.

[20] R.R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference*, pages 731–736. ACM, 2010.

[21] D. Riley, E. Eyisi, J. Bai, X. Koutsoukos, Y. Xue, and J. Sztipanovits. Networked control system wind tunnel (ncswt)-an evaluation tool for networked multi-agent systems. *Proc. of SIMUTools*, 2011.

[22] L.A. Rossman, Center for Environmental Research Information, Estats Units d'Amèrica. Environmental Protection Agency. Office of Research, Development, and National Risk Management Research Laboratory. Epanet 2: users manual. 2000.

[23] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang. Cyber-physical systems: a new frontier. *Machine Learning in Cyber Trust*, pages 3–13, 2009.

[24] T. Stahl and M. Volter. Model-driven software development. *Technology, Engineering, Management. England: John Wiley & Sons*, 2006.

[25] J.D. Stefanovski and G.M. Dimirovski. Automated open-canal water supply systems. Automatic systems for building the infrastructure in developing countries. *Elsevier Science Ltd.*, pages 189–194, 2001.

[26] R. Szymkiewicz. Numerical modeling in open channel hydraulics (Water science & technology library). 2010.

[27] M.U. Tariq, S. Grijalva, and M. Wolf. Towards a distributed, service-oriented control infrastructure for smart grid. In *Cyber-Physical Systems (ICCPS), 2011 IEEE/ACM International Conference on*, pages 35–44. IEEE, 2011.

[28] W. Wolf. Cyber-physical systems. *Computer*, pages 88–89, 2009.